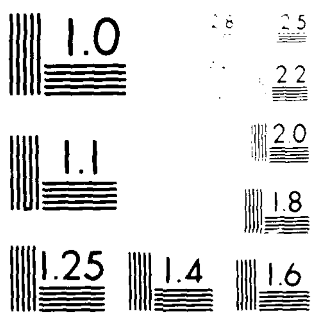
AD-A110 322 DEFT LABS INC EAST SYRACUSE NY F/G 20/1
ACOUSTO-OPTIC TECHNOLOGY FOR TOPOGRAPHIC FEATURE EXTRACTION AND--ETC(U)
MAR 81 A L MOYER DAAK70-79-C-0160
UNCLASSIFIED 0102-A002 ETL-0256 NL

1 OF 3
AD A
110322


1.0
2.5
2.2
1.1
2.0
1.8
1.25
1.4
1.6

AD A110322

ETL-0256

LEVEL II

Acousto-optic technology for topographic feature extraction and image analysis

Alan L. Moyer

Deft Laboratories, Inc.
7 Adler Drive
East Syracuse, NY 13057

MARCH 1981

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

U.S. ARMY CORPS OF ENGINEERS
ENGINEER TOPOGRAPHIC LABORATORIES
FORT BELVOIR, VIRGINIA 22060

E T L

U.S. ARMY ENGINEER TOPOGRAPHIC LABORATORIES

ETL

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER ETL-0256	2. GOVT ACCESSION NO. AD-A110 331	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) ACOUSTO-OPTIC TECHNOLOGY FOR TOPOGRAPHIC FEATURE EXTRACTION AND IMAGE ANALYSIS		5. TYPE OF REPORT & PERIOD COVERED Final Technical Report Oct.1979 thru Feb. 1981
		6. PERFORMING ORG. REPORT NUMBER 0102-A002
7. AUTHOR(s) Alan L. Moyer		8. CONTRACT OR GRANT NUMBER(s) DAAK70-79-C-0160
9. PERFORMING ORGANIZATION NAME AND ADDRESS Deft Laboratories, Inc. 7 Adler Drive East Syracuse, NY 13057		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS U.S. Army Engineer Topographic Laboratories Fort Belvoir, VA 22060		12. REPORT DATE March 1981
		13. NUMBER OF PAGES 205
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Image Processing, Acousto-Optics, Algorithms, Feature Extraction		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) This report contains all findings of the acousto-optic technology study for feature extraction conducted by Deft Laboratories Inc. for the U.S. Army Engineer Topographic Laboratories. The objective of this program was to develop, analyze and evaluate theoretical concepts and strategies for topographic feature extraction and image analysis using acousto-optic (A-O) technology.		

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

412038

20. Abstract (cont'd.)

A conclusion of this study was that A-O devices are potentially capable of implementing the feature extraction prefilter function very efficiently. Since the prefilter is the most computational intensive portion of the feature extraction process this is a significant result. The best application of A-O devices is in the implementation of transform-based prefilter algorithms. Under this contract transform-based algorithms were identified and developed which are invariant to feature translation, rotation and scale. This invariance is highly desirable since it reduces the number of distinct feature signatures which must be processed by the decision processor.

Some preliminary experiments were conducted using the Fourier-based algorithms, test images and A-O device which was a Deft sensor. This combination of algorithms and sensor was able to distinguish between three test patterns which were presented in arbitrary orientation and scale. The success rate was 80%. In spite of these promising results, present Deft sensors are not capable of distinguishing realistic features in aerial photographs. New Deft sensors, presently under development, are described which will significantly improve the capability of this sensor in feature extraction applications.

PREFACE

This report contains all findings of the acousto-optic technology study for feature extraction conducted by Deft Laboratories Inc. for the U.S. Army Engineer Topographic Laboratories. The work reported here was funded under Contract DAAK 70-79-C-0160. The work was conducted during the period of October 1979 through February 1981. The Contracting Office's Technical Representative was Mr. Joseph F. Hannigan.

SUMMARY

The objective of this program was to develop, analyze and evaluate theoretical concepts and strategies for topographic feature extraction and image analysis using acousto-optic (A-O) technology.

A conclusion of this study was that A-O devices are potentially capable of implementing the feature extraction prefilter function very efficiently. Since the prefilter is the most computational intensive portion of the feature extraction process this is a significant result. The best application of A-O devices is in the implementation of transform-based prefilter algorithms. Under this contract transform-based algorithms were identified and developed which are invariant to feature translation, rotation and scale. This invariance is highly desirable since it reduces the number of distinct feature signatures which must be processed by the decision processor.

Some preliminary experiments were conducted using the Fourier-based algorithms, test images and an A-O device which was a Deft sensor. This combination of algorithms and sensor was able to distinguish between three test patterns which were presented in arbitrary orientation and scale. The success rate was 80%. In spite of these promising results, present Deft sensors are not capable of distinguishing realistic features in aerial photographs. New Deft sensors are presently under development which will significantly improve the capability of this sensor in feature extraction applications.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF FIGURES - Cont'd.

LIST OF TABLES

I. INTRODUCTION

A. Project Overview

The objective of this program was to develop, analyze and evaluate theoretical concepts and strategies for topographic feature extraction and image analysis using acousto-optic (A-O) technology. In addition, functional block diagrams were to be prepared for the most promising concepts and strategies for the purpose of identifying essential and/or critical A-O elements and components.

During the execution of the program it became evident that the evaluation of theoretical concepts was sufficiently advanced so that additional effort, not required in the original contract, could be undertaken. This effort involved programming a group of promising feature extraction algorithms and performing experiments using an A-O device. The device used in these experiments was a Deft sensor.

The tasks which were executed during this program are listed below.

1. The open literature was searched in the areas of image processing, feature extraction and acousto-optic technology.
2. A survey was conducted and determination made of device capability and limitations of present A-O technology.
3. Theoretical concepts and strategies were analyzed and evaluated for topographic feature extraction and for implementation using A-O technology.
4. The most promising concepts and strategies were selected for further evaluation.
5. For these strategies algorithms were developed and programmed for a microprocessor-based experimental system. Experiments were conducted using this system and a Deft sensor. These experiments involved feature recognition using a small set of test images.
6. Functional block diagrams were developed for the most

promising strategies. Essential and/or critical A-O elements were identified.

7. For these A-O elements state-of-the-art capabilities were compared with required capabilities. Required improvements were determined as well as indication of the probability of obtaining such improvements.

This report details all findings of this study program.

B. General Feature Extraction System Model

The term "feature extraction" comes under the heading of pattern recognition. Large volumes of material have been written under this heading. The specific problem addressed by this study is how can A-O devices be used to find objects or "features" in aerial photographs? The emphasis, then, is on what functions A-O devices can perform and how these functions can be used to achieve the stated objective. In this context, only certain portions of the literature under pattern recognition has bearing on the problem to be solved.

It was felt at the beginning of this study that the only way to achieve meaningful results was to determine a model for a feature extraction system which has the general functional capability required. From this general model, more specific systems could be developed which utilize functions which can be efficiently performed by A-O devices. Concepts and strategies could then be chosen from the literature under pattern recognition which fit into these specific system models.

In order to provide a frame of reference, feature extraction will be considered in the context of the pattern-recognition system model shown in Figure 1. The transducer transforms the information content of the input (photolight intensity pattern) into a format suitable for further processing (electrical signal). The amount of information available in the photo image is enormous. A single video image may contain 10^4 - 10^8 bits of

information, while a photographic image can contain orders of magnitude more. However, the amount of information required to make a decision may be just a few tens of numbers. The purpose of the preprocessor is to reduce the information content of the image to a more manageable size. This process has also been termed filtering, prefiltering, feature or measurement extraction or dimensionality reduction. The output of the preprocessor is then passed to the decision processor where this information is used to classify the photo image. The decision processor may also control the transducer and preprocessor via feedback. In this manner the preprocessor function may be changed during the decision process. In like manner the transducer may be commanded to look at a different portion of the photo or perhaps change scale.

Figure 1 is a functional block diagram. These functions may be distributed over hardware subsystems in a number of ways. For example, the transducer might be a vidicon or CCD imager. Its function then is primarily bandwidth reduction and format conversion. The preprocessor and decision processor might then be implemented in software in a digital computer.

Another alternative configuration is considered in some detail in this report. In this configuration the transducer and preprocessor functions are performed in an acousto-optic device. The decision process is then carried out in a digital processor. In an all digital implementation the preprocessor function requires the bulk of the processing time since a large data base (image) must be operated on. The potential advantage of the acousto-optic implementation is that this time-consuming preprocessing function can be performed in whole or in part in the sensor itself.

A third configuration should be mentioned. In this configuration the transducer and preprocessor functions are performed by an optical processor. Many of the prefilter functions which can be computed in an optical processor can also be computed by acousto-

optic devices. As a result, the large literature which deals with applications of optical processors can be utilized to develop preprocessing techniques for acousto-optic devices. The development of both optical processors and acousto-optic devices is currently receiving support because of their potential application in such areas as image preprocessing. At this time, optical processors are in a more advanced state of development than acousto-optic devices. However, acousto-optic devices offer the potential advantages of being cheaper, more rugged and not requiring precise optical alignment when compared to optical processors. In certain applications they may also prove to be more flexible since the function performed can be controlled electrically.

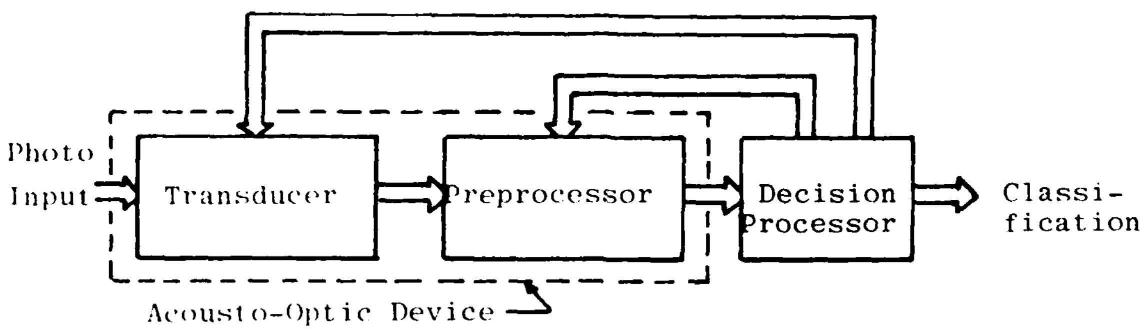


Figure 1 - Pattern Recognition System Model[1]

II. ACOUSTO-OPTIC DEVICES

A. Introduction

In this section three acousto-optic devices are briefly described. Both physical and functional descriptions are provided. From these descriptions it will then be possible to determine which feature extraction algorithms are best suited for implementation using one or more of these devices. More detailed descriptions can be found in the references. The devices to be described are the following: the Deft sensor manufactured by Deft Laboratories Inc., an elastobirefringent light valve and a device developed by Thomson-CSF, France. Both the Deft sensor and the elastobirefringent light valve were developed by Drs. Kowel and Kornreich of Deft Laboratories. As a result we are more familiar with the present limitations and future potential of these devices. Section V is devoted to a discussion of this topic.

B. The Deft Sensor

The Deft sensor is a solid state device which utilizes the interaction of surface acoustic waves (SAW), a photoconducting film and an imaged light pattern to produce an electrical signal from which can be derived the magnitude and phase of the two-dimensional Fourier transform of the image pattern. The sensor may also be used to produce other useful image functions.

The operation of the sensor can be explained with the aid of Figure 2. The sensor is fabricated on a $LiNbO_3$ substrate which is a piezoelectric material. On this substrate is deposited a photoconducting film of CdS. An interdigital metal pattern is evaporated onto the CdS for the purpose of detecting and integrating the photocurrent. This pattern has the added function of sampling the image in one direction. Image sampling is required if base-band Fourier transform components are desired. The SAW are limited to high spatial frequencies. The spatial spectrum of the image must be translated into this band. To achieve a

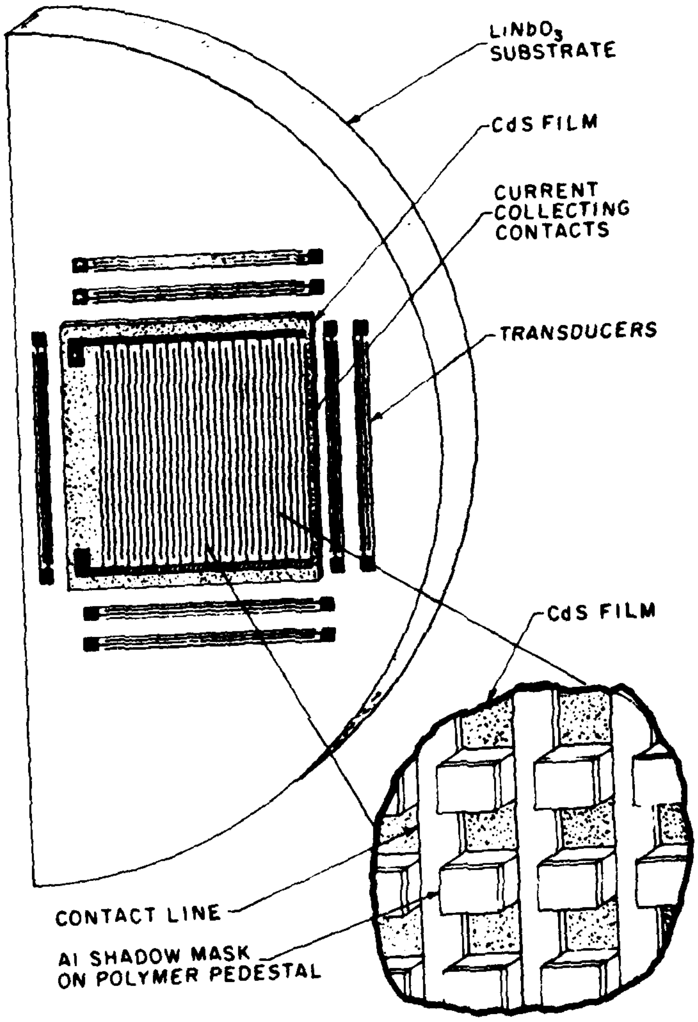


Figure 2 - Deft Sensor Construction

spatial sampling in the orthogonal direction the CdS can be overlayed with an Al shadow mask. As an alternative which has been used in more recent devices, only squares of CdS with the proper sample spacing are deposited. Interdigital transducers are used to generate two orthogonal surface acoustic waves in the substrate. The figure shows a number of transducers. However, only two are used to launch the orthogonal SAW's. The other transducers are redundant and not used.

To describe the operation of the sensor let g(t) and h(t) be the transducer input signals. Since $LiNbO_3$ is piezoelectric an electrical field is produced across the sensor. In the x direction the field is proportional to $g(t-x/v_x)$. In the y direction the field is proportional to $h(t-y/v_y)$. The parameters v_x and v_y are the SAW velocities of propagation in the x and y directions. At the same time, an image is focused on the grid of CdS squares. A current is generated in each square proportional to the average light intensity over that square, and these currents are modulated by the product, $g(t-x/v_x)\ h(t-y/v_y)$, created by nonlinear mixing in the CdS film. The metal finger pattern sums each of the modulated current components over the grid. The resulting output current is o(t). It is given by

$$o(t) = \iint \hat{I}(x,y) g(t-x/v_x) h(t-y/v_y) dx dy \qquad (1)$$

The image $\hat{I}(x,y)$ is the image focused on the sensor, I(x,y) modified by the grid sampling. The functions g(t), h(t) can be any arbitrary waveforms who's spectrum is within that of the interdigital transducers. This equation defines the function of the Deft sensor in its most general form.

A number of specific functions which can be realized by the sensor may have application in feature extraction. These will be discussed in the next section. A specific function which may have application in feature extraction is the two-dimensional Fourier transform. Since the performance of the Deft sensor in general applications can best be described in terms of its spectral response, computation of the two-dimensional Fourier transform using the sensor will be described here.

If g(t) and h(t) are sinusoids then o(t) yields components of the two-dimensional Fourier transform.

$$F(\omega_x, \omega_y) = \iint I(x,y) e^{-j(\omega_x x + \omega_y y)}\, dxdy$$

By varying the frequencies of the sinusoids different spatial frequencies can be probed. Let ℓ_x, ℓ_y be the grid spacing in the x and y direction respectively in meters/cycle. Then the temporal frequency which corresponds to the origin of the spatial transform is given by $f_{o,x}$, $f_{o,y}$ where

$$f_{o,x} = v_x/\ell_x, \quad f_{o,y} = v_y/\ell_y \qquad (2)$$

For example, if ℓ_x = 0.1 mm/cycle, v_x = 3800 m/s then $f_{o,x}$ = 38 MHz.

The number of spatial frequencies which can be probed by the sensor depends on the bandwidths of the two SAW transducers and the size of the CdS grid area. Let n_x and n_y be the number of CdS squares in the x and y direction respectively. Then for two spatial frequencies to be resolvable by the sensor, the total number of cycles of one frequency across the sensor must differ from the total number of cycles of the other by at least one cycle. For example, consider two x-directed spatial frequencies which differ by one cycle across the sensor. If n_x is the number of squares of CdS in the x direction then the difference in spatial frequency is $1/(n_x \ell_x)$. The difference in temporal frequency is then $v_x/(n_x \ell_x)$. Comparing this to equation (2) shows that the difference frequency, Δf_x is

$$\Delta f_x = f_{o,x}/n_x \qquad (3)$$

Now let ∇f_x be the temporal bandwidth of the x transducer. Then the number of resolvable frequencies in the x-direction is

$$r_x = \frac{\nabla f_x}{\Delta f_x} = \frac{\nabla f_x}{f_{o,x}/n_x} = n_x \frac{\nabla f_x}{f_{o,x}} \qquad (4)$$

That is, the number of resolvable components in the x direction is n_x times the percentage bandwidth of the x transducer. For example, if $\nabla f_x/f_{o,x}$ = 0.1 and n_x = 200 then there will be 20

resolvable Fourier components in the x direction. For $\ell_x = 0.1$ mm/cycle this corresponds to a length of 20mm for the CdS portion of the sensor. In like manner, the number of resolvable frequencies in the y direction is given by

$$r_y = n_y \frac{\nabla f_y}{f_{o,y}} \tag{5}$$

The number of resolvable Fourier components is an important parameter in applications such as feature extraction. It is used to characterize the operation of the Deft sensor even when a function other than the Fourier transform is being computed. For example, consider the more general function

$$u = \int\int \hat{I}(x,y)g(x)h(y)\ dxdy \tag{6}$$

to be computed by the sensor. Because of the limits of integration g and h can be replaced by their periodic extensions $[g]$ and $[h]$. These functions will be defined by an example. Assume that $g(x) = x$ and that the x limits of integration are from -1 to +1. Then the periodic extension of $g(x)$ is shown in Figure 3.

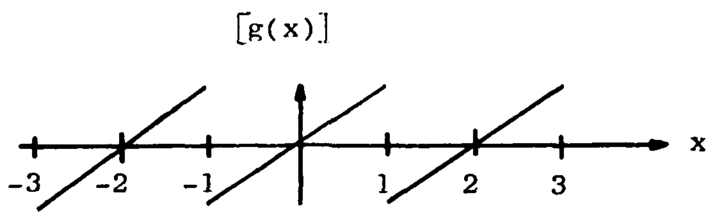


Figure 3 - Periodic Extension of $g(x) = x$

The function $[h(y)]$ is likewise defined with respect to the limits of integration in the y direction. Because $[g]$ and $[h]$ are periodic they may be expanded in complex Fourier series.

$$[g] = \sum_{k=-\infty}^{\infty} c_{g,k}\ e^{(jk\pi x/2)} \tag{7}$$

$$[h] = \sum_{k=-\infty}^{\infty} c_{h,k} \; e^{(jk\pi y/2)} \tag{8}$$

where $j = \sqrt{-1}$. Substituting these series in equation (6) and interchanging the order of integration and summation yields

$$u = \sum_{i=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} c_{g,i} \; c_{h,k} \iint \hat{I}(x,y) \; e^{j\pi(ix + ky)/2} \, dxdy \tag{9}$$

$$= \sum_{i=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} c_{g,i} \; c_{h,k} \; F(i\pi/2, \; k\pi/2) \tag{10}$$

where F is the two-dimensional Fourier transform of $\hat{I}$. Since the Deft sensor transducer bandwidths are limited, u can be approximated by the truncated series

$$\hat{u} = \sum_{i=-n_x}^{n_x} \sum_{k=-n_y}^{n_y} c_{g,i} \; c_{h,k} \; F(i\pi/2, \; k\pi/2) \tag{11}$$

where n_x+1, n_y+1 are the number of resolvable Fourier components along the ω_x and ω_y axis respectively. The error in computing u with the sensor is

$$u - \hat{u} = \sum_{|i|>n_x} \sum_{|k|>n_y} c_{g,i} \; c_{h,k} \; F(i\pi/2, \; k\pi/2) \tag{12}$$

This error depends on n_x, n_y and on the high frequency content of the image $\hat{I}$ and of the kernal $g(x)h(y)$. Hence, given a sensor with parameters n_x, n_y it is possible to determine which kinds of images may be viewed and which kinds of kernals may be utilized.

The Deft sensor is further described in a number of references [2,3,4].

C. Elastobirefringent Light Valve

A two-dimensional acoustic processor utilizing an elasto-birefringent light valve has been constructed and experiments conducted [5,6]. The experimental setup is shown in Figure 4. The light valve consists of a fused quartz cell. On the edges of this cell are attached two orthogonal transducers. When

these transducers are excited strain waves travel through the bulk of the cell. The total strain is the linear superposition of the two traveling strain waves. An image is focused on the region of the quartz where the two waves are superimposed.

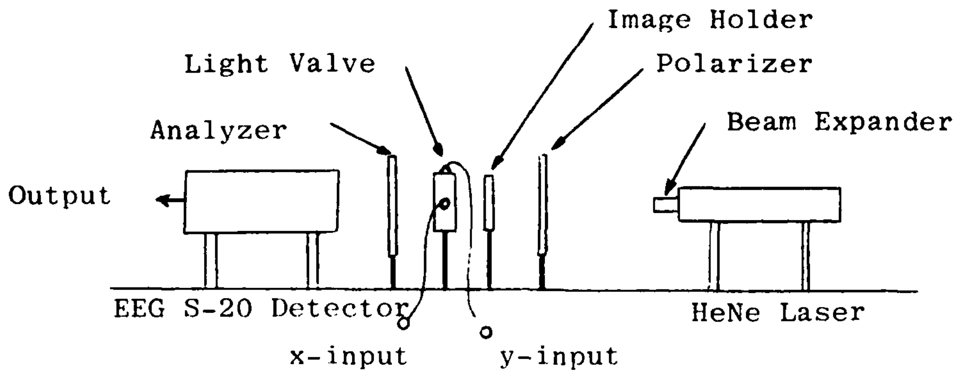


Figure 4 - Elastobirefringent Light Valve: Experimental Set-Up

Due to the strain induced birefringence, each ray of light splits into two orthogonally polarized phase velocities. It can be shown that the intensity modulation of each polarized wave has a component proportional to the strain squared. The image is produced by illuminating a transparency containing the image information with a well-collimated beam which has been polarized. The output of the light valve will have a desired component polarized orthogonal to the orientation of the input polarizer. This component is passed by the analyzer which is just another polarizer. The undesirable direct component is blocked by the analyzer. The desired component is then detected by a photodetector which performs an integration over the entire image. A chromium sampling grid is placed on the quartz cell to sample the image and translate its spatial spectrum into the passband of the transducers.

The function which can be computed by this processor can be described by equation (1) which is also used to describe the

Deft sensor. Hence, functionally this processor is equivalent to the Deft sensor. The differences between the two are primarily in implementation and in present and projected future performance. Concerning implementation, the light valve is a larger physical structure than the Deft sensor. The image source must be a transparency. By contrast, the Deft sensor can view either a transparency, a photograph or a real-world image since it operates much like a conventional camera. The elastobirefringent light valves which have been built to date exhibit a strong standing wave pattern which is a function of transducer frequency. This effect has limited their usefulness. Potential solutions to this problem are presented in Section V.

This light valve does have a couple of potential advantages compared with the Deft sensor. In the bulk mode devices it is not necessary to physically attach a metal pick-up grid. In the Deft sensor this grid has an undesirable effect of damping the SAW. This limits the practical size of Deft sensors since larger sensors would suffer a large, undesirable SAW attenuation across the sensor surface. Hence, the bulk devices could potentially be larger resulting in a greater number of resolvable Fourier components. A second advantage is that electrical feed-through is not a problem with the bulk devices. With the Deft sensor feed-through from the input transducer drive to the integrating metal grid is a problem.

However, the bulk device has a number of disadvantages when compared with the Deft sensor. Since bulk mode operation is required, the image must be focused with sufficient depth-of-focus so that the image is in focus throughout the 1-2mm thickness of the quartz cell. Higher acoustic fields are required to operate the bulk device. The light valves built to date use glued-on transducers. This is not desirable, particularly at higher frequencies where the transducers are smaller and more brittle. Finally, the physical non-linearity which is utilized in the bulk light valve is a much smaller effect than

that used in the Deft sensor. There is a large, undesirable unmodulated signal which feeds through the light valve. The purpose of the analyzer is to block this signal. However, it is difficult to fabricate sheet polarizers which are uniform enough over the 1 cm. illuminated length of the quartz cell. Hence, some of this feed-through signal will pass the analyzer and could saturate the detector. (This feed-through signal is analogous to the electrical feed-through encountered in the Deft sensor.)

D. Thomson - CSF Sensor

The Thomson - CSF sensor is a SAW device and has features in common with the Deft sensor. The device can be described as a two-dimensional separated media semiconductor convolver. It employes two pairs of SAW transducers deposited on a $LiNbO_3$ substrate. An 8 x 8 mm^2 matrix of p-n vidicon type diodes is pressed on 3000 A high randomly distributed posts. Output signals are picked up between the back electrode of the semiconductor and a semi-transparent ground electrode deposited on the bottom face of the piezoelectric medium. The device is shown in Figure 5. The x and y center frequencies are $f_{o,x}$ and $f_{o,y}$. The useful output frequency then appears at frequency $2(f_{o,x} + f_{o,y})$.

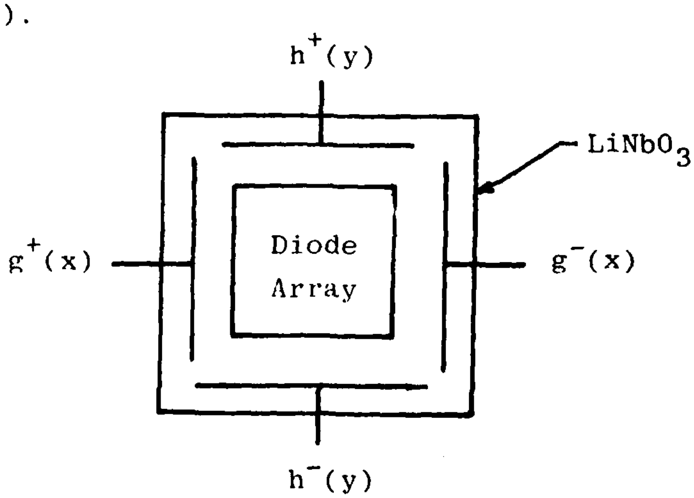


Figure 5 - Thomson - CSF Sensor

To generate the two-dimensional Fourier transform of an image focused on the diode array, the x input signals are chosen to be FM ramps of opposite slopes $\omega_x^{\pm} = \omega_x \pm ut$. The y input signals are chosen to be CW signals with frequencies $\omega_y^{\pm} = \omega_y \pm 1/2\, \Delta\omega_y$. (That is, one x orientated transducer gets ω_x+ and the other x orientated transducer gets ω_x^-. The same is true for y orientated transducers.) The instantaneous frequencies of these waveforms are shown in Figure 6 (solid lines).

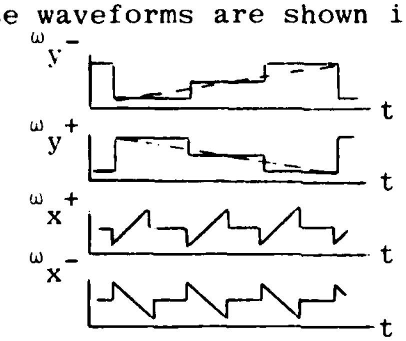


Figure 6 - Transducer Instantaneous Frequencies

With these inputs, the output signal component at $2(\omega_x + \omega_y)$ can be shown to have a modulation of the form

$$o(t) = \iint \hat{I}\,(x,y)\; e^{j(-2utx/v_x + \Delta\omega_y/v_y)}\, dxdy \tag{13}$$

where v_x, v_y are the SAW velocities of propagation. During a sweep of the x, FM ramps an entire row of the transform is read from the sensor. This sweep time need be no more than twice the propagation time across the sensor. Hence, data can be outputed potentially much faster than can be achieved using the Deft sensor. In the Deft sensor the x and y input waves must be CW signals which must propagate across the sensor before a single transform component is outputted. The frequencies of these waves are then stepped to output another component, etc. The Thomson - CSF sensor approach would be most useful if a raster-scan of the transform is desired. This is shown in Figure 7 (solid lines).

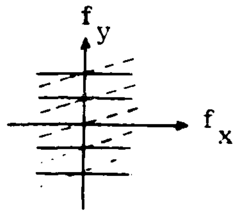


Figure 7 - Raster Scan of Transform Plane

If instead of stepping the y frequencies, the y frequencies are also ramps (dashed lines, Figure 6) then the transformed plane can also be raster-scanned along oblique lines with variable tilt angle (Figure 7, dashed lines). For example, it would be possible to scan the transform along radial lines passing through the transform origin. This is shown in Figure 8.

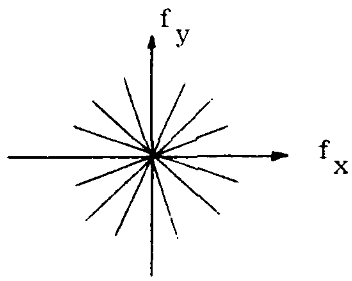


Figure 8 - Radial-Scanned Transform

This method of scanning will be shown in Section III to have application in feature extraction.

The primary difference between the Deft sensor and the Thomson - CSF sensor is the use of four vs. two transducers and the choice of waveforms leading to a fast scan of the transform. If random samples or samples at only a few spatial frequencies are all that is required then the Deft sensor would function just as rapidly as the Thomson - CSF sensor. It would be possible to operate a Deft sensor in the same manner as the Thomson - CSF sensor by also using four transducers and inputting FM ramp waveforms. Speed performance would then

equal the Thomson - CSF device. This modification has not been attempted to date. Note that the method used to speed-up data output is not applicable to the computation of the more general operator of equation (1).

The Thomson - CSF sensor is described in reference 7.

III. FEATURE EXTRACTION TECHNIQUES FOR ACOUSTO-OPTIC TECHNOLOGY

A. Introduction

During the execution of this contract a survey of techniques for feature extraction was carried out. The purpose of this survey was to determine techniques suitable for implementation using acousto-optic devices. The results of this survey are detailed in this chapter. From the techniques surveyed, two methods appear to offer promise both from an algorithmic standpoint and also because of suitability of implementation. These methods will be termed the method of invariant Fourier signatures (IFS) and the method of invariant moment signatures (IMS). Because of their promise, additional algorithmic development was carried out during the course of this contract. This work is reported in this section.

This section is divided into two parts. The first part is a general discussion of feature extraction. The second part is a comparison of some methods for feature extraction.

B. General Concepts

A general model for a feature extraction system is presented in Section 1B. This model is shown in Figure 1. Briefly, the transducer converts the image input light pattern to electrical signals. The preprocessor reduces the information content of the image to a more manageable size. The output of the preprocessor is then passed to the decision processor where this information is used to classify the photo image.

In Section II the operation and function of three A-O devices were considered. The general function performed by these devices is given by equation (1). This equation implies that the two-dimensional image information is processed by the sensor in such a way to result in a one-dimensional signal of reduced complexity. Hence, these devices could be potentially used to implement the combined function of transducer and preprocessor. Therefore, it is important to consider specific

examples of equation (1) and determine their utility in preprocessing. This will be done in this section.

The A-O devices considered do not seem useful for implementing the decision processor. There are a number of reasons for this. First of all, the output of the preprocessor will be an electrical signal. The A-O devices require an image input. Hence, the data format is not compatible. A survey of the types of algorithms commonly used for classification reveals that most of these cannot be implemented efficiently using A-O devices. In addition, they require greater accuracy than can be achieved with A-O devices. But most importantly, it is not necessary to use A-O devices for classification since the image information content has been reduced by the preprocessor to the point where a modest digital processor can handle this function. A survey of conventional classification methods is given in reference 8. An application of classification in aircraft identification is given in reference 9.

Before developing specific preprocessor functions some desirable properties of the transducer plus preprocessor will be discussed.

The first desirable property is feature isolation. Consider a typical aerial image of natural terrain in which may be located one or more man-made features to be detected. The natural terrain can be considered to be "noise" while the feature is a "signal". One of the functions of the preprocessor is to filter this signal from the noise. Now in a typical aerial photograph, most of the photo will be noise. If the transducer were to view the entire photograph then the signal-to-noise ratio would be small and the preprocessor function would be more difficult to implement. A simple but effective way to improve the signal-to-noise ratio would be to limit the field of view of the transducer to only a portion of the photo and then scan the photo to search for features. Limiting the field of view has the effect of reducing the noise without reducing the signal as long as the feature to be detected remains completely within the field of view.

By scanning the photo is meant that the transducer views a portion of the photo and that portion is preprocessed. The transducer then views another portion of the photo and preprocessing is repeated for that portion. This sequence of steps is continued until all portions of the image have been viewed. Scanning may or may not involve overlapping views. Scanning a square photo with a square aperture and no overlapping is shown in Figure 9. The same scan but with 50% overlap in both directions is shown in Figure 10. To simplify presentation the views for the overlap case are shown in four parts. These parts would overlap each other. The boundary of the photo is shown dashed. Now consider a feature which is completely enclosed by the circle shown in Figure 9. (The angular orientation of the feature is arbitrary.) Then if the photo was scanned as in Figure 9 the feature would lie across the boundary of four views and its detection would be difficult. However, if 50% overlapping of views is used as in Figure 10 then the feature lies totally within one view (in this case, view 48). As long as the radius of the circle was less than or equal to one quarter of the length of the square comprising each view and overlap was 50% the circle would lie completely within one view. Note that (neglecting missing squares at the edges of the photo) the number of views has increased by a factor of four.

These comments can be generalized. Let r be the radius of the smallest circle which completely encloses the feature. Obviously, r is independent of the angular orientation of the feature in the plane of the image. Assume that the aperture of the transducer imaged on the photo is square with length ℓ. Figure 11 shows one such aperture (large square). In the middle of this square is a smaller square. The area enclosed by this square is the locus of points of all circles of radius r which lie completely within the large square. Each view has such a small square associated with it. In order that the

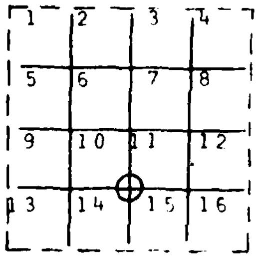


Figure 9 - Image Scanning: No Overlap

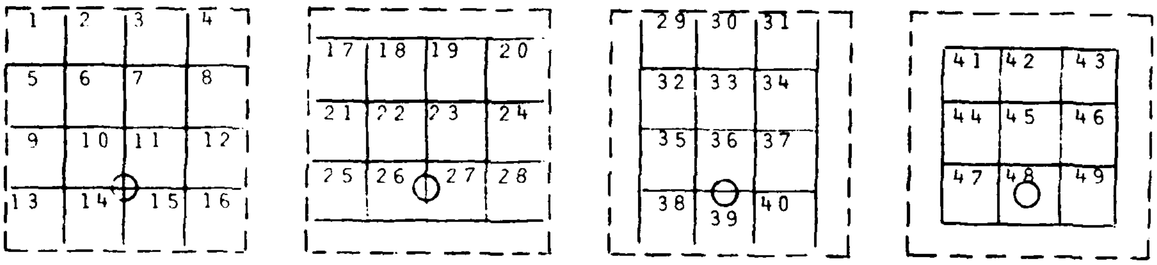


Figure 10 - Image Scanning: 50% Overlap

feature lie totally within a view regardless of the position of the feature within the image, the views must overlap in such a way that the entire area of the photo is covered by the union of all these small squares. In that case, the total number of views must equal at least

$$\text{total number of views} = \ell_T^{\,2}/(\ell - 2r)^2 \tag{14}$$

where ℓ_T is the length of the square photo. That is, the total number if views is greater or equal to the total area of the photo divided by the area of a square of length $\ell - 2r$.

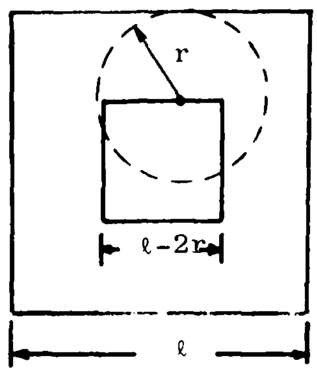


Figure 11 - Overlap Geometry

The signal-to-noise ratio (SNR) of the view will be defined to equal the area of the feature divided by the area of the view, Hence,

$$\text{SNR} \leq 2\pi r^2/\ell^2 \tag{15}$$

with maximum occuring when $\ell = 2r$. However, in that case the total number of views required is infinite. Hence, there must be a tradeoff between SNR and total number of views to be processed. If the ratio r/ℓ_T is very small (small features) then a large number of views are required to achieve good SNR. For example, assume that $2r = \ell_T/50$ and that $\ell = 12r$ then SNR $\leq \pi/72$ and the total number of views is 100.

In practice, the total number of views can be reduced somewhat by allowing some of the feature to be masked by the

transducer aperture. That is, r is chosen to be somewhat smaller than the value implied from the dimensions of the feature. To develop a general formula let $\ell = k_1 r$. Now reduce r to $k_2 r$. Then the total number of views is reduced by a factor of $(k_1-1)^2/(k_1-k_2)^2$. In the previous example k_1 = 12. Let $k_2 = 0.8$. That is, overlap will be reduced so that, in the worst case, only 80% of the largest feature dimension will be within a view. In this case the total number of views required is reduced by a factor of 0.9646. For a fixed value of k_2 this reduction will be greater for smaller values of k_1 >2. For example, if $k_1 = 4$ and $k_2 = 0.8$ then the total number of views is reduced by a factor of 0.8789.

In terms of implementation, a single transducer and prefilter could be used to scan the photo in a number of steps. If high-speed operation was required more than one transducer and prefilter could view seperate views of the photo concurrently.

Another desirable property of the prefilter will now be discussed. By definition, the prefilter takes the large amount of information of the photo encoded as pixels and reduces it to a much smaller set of numbers. This process can be termed "dimensionality reduction". Ideally, the prefilter output should have the following properties:

1) Members of each feature class should show less variability than was the case before the prefilter.

2) The relative separation of each feature class should be increased.

3) There should be dimensionality reduction.

Consider first property 1). As an example of a feature class let the class include all square buildings in the photo. In this case, a square building is in the feature class regardless of its angular orientation or scale (size). It would be desirable for the prefilter to have the property that if the photo contains a square building in any position, orientation or scale then the prefilter output will contain a signal or

signature which is invariant to position, orientation or scale. Such a prefilter will be said to implement a position, rotation and scale invariant transformation. In certain cases it may be desirable that the transformation is only position and rotation invariant. (Example: small buildings are to be discriminated from large buildings.)

The best-known invariant transformation is the autocorrelation function. The first-order autocorrelation function defined in one dimension is

$$\phi_g^1(\Delta) = \int g(x)g(x+\Delta)\, dx \qquad (16)$$

this function is invariant to translation.

Consider now property 2). As an example, assume that the prefilter input is a photo containing a square building. This building may be in any position, orientation and scale. The set of all outputs constitutes a class at the prefilter output. In this idealized situation, if the prefilter is invariant to translation, rotation and scale then this class contains but one output or signature. A more realistic consideration would have to include such effects as variation of building materials and color and lighting. Another class could be determined by considering all outputs for all possible orientations of "round storage tanks" in a photo. Now if the differences in all signatures of one class from all signatures in the other class is increased by prefiltering then relative separation has increased. This might be the case, for example, if noise, such as natural features, is filtered out by the prefilter.

However, in some cases relative separation may decrease. For example, consider the autocorrelation function. It is well-known that

$$F\{\phi\} = |F\{g\}|^2 \qquad (17)$$

where F{} is the Fourier transform. Hence, two functions who's Fourier transforms have the same magnitude cannot be separated by using the autocorrelation functions. Functions which cannot

be separated include not only those which differ by a linear phase term (i.e., the translated versions of the same function) but also those which differ by a nonlinear phase term.

A function which avoids this difficulty is the higher-order autocorrelation function

$$\phi^n_g(\Delta_1, \Delta_2, \cdots \Delta_n) = \int g(x)g(x+\Delta_1)g(x+\Delta_2)\cdots g(x+\Delta_n)\, dx \quad (18)$$

which can be shown to be unique for $n \geq 2$ except for translation[10]. However, note that the dimensionality has been increased. This conflicts with the third desirable property that there should be a dimensionality reduction.

In general, it is not possible to satisfy all three desirable features of a prefilter at the same time. It would seem to be important that there be a dimensionality reduction since, otherwise, the amount of information presented to the decision processor would be enormous.

Later in this chapter two classes of algorithms are presented which satisfy properties 1) and 3) but violate property 2). This would seem to be the lesser evil since the decision processor receives less information which has been formatted in such a way that the decision process is simplified. Since property 2) is violated, a square building, for example, may be mistaken for a round storage tank. However, this is less important than not detecting the feature. It is always possible to refine the decision process by human intervention. A feature extractor which signals the presence of a feature and indicates with some certainty what that feature is could be very useful for automatic aerial photo screening.

C. Transform Methods

Because of the scope of this project, this section will concentrate on prefilters which utilize transforms which can be implemented as a special case of equation (1). Transforms which will be considered are the two-dimensional Fourier transform, the two-dimensional image moments and the two-

dimensional Hadamard transform. Applications of the Fourier and Hadamard transform to image processing is surveyed by Pickholtz[11]. The application of moments to image processing is discussed in a number of papers [9,12,13].

1. Hadamard Transform

The Hadamard transform of an image which has been sampled and represented by the nxn matrix of pixel values $I(x_i, y_j)$ is

$$F(i,j) = \frac{1}{n} H_n I H_n \tag{19}$$

the nxn matrix H_n consists of elements which are either +1 or -1. The Hadamard transform is orthonormal. As a result, the image is decomposed into basis images which are the two-dimensional Walsh functions. The rows and columns of the matrix H_n (which is symmetric) consists of the one-dimensional Walsh functions of order n. For n = 8 these Walsh functions are shown in Figure 12.

		Sequency
wal (o,t)	++++++++	0
wal (1,t)	++++----	1
wal (2,t)	++----++	1
wal (3,t)	++--++--	2
wal (4,t)	+--++--+	2
wal (5,t)	+--+-++-	3
wal (6,t)	+-+--+-+	3
wal (7,t)	+-+-+-+-	4

Figure 12 - First Eight Walsh Functions

The notation + means +1 while - means -1. These functions are defined on the interval $0 \leq t \leq 1$ and may be periodically extended to span the real line. The Walsh functions may be thought of as square waves. Wal (6,t) is shown in Figure 13 a). For each of these functions, every occurrance of a transition from + to - or from - to + is called a zero crossing. One half the number of zero crossings of these functions is termed the sequency of the function. Sequency is analogous to frequency of sine and cosine functions.

The transform $F(i,j)$ is a decomposition of the image into its orthonormal Walsh basis images. The potential advantage of this representation is that it may be possible to accurately approximate the image by using only a limited number of its Hadamard transform components. If so, this would facilitate feature extraction since the output of the Hadamard prefilter would be of lower dimensionality than the input.

The Hadamard transform also appears useful because of its ease of implementation. Because H_n only consists of +1 or -1 entries the matrix product represented by equation (19) can be computed without any multiplies. This is particularly useful in a digital processor where multiplications require more computation than do adds and subtracts. A high-speed digital processor which computes the Hadamard transform of an image for feature extraction is described in reference 14.

One potential disadvantage of the Hadamard transform is that it is not invariant to translation, rotation or scale of a feature in the image.

Consider now the computation of the Hadamard transform using acousto-optic devices. The potential application of the Deft sensor to computing the Hadamard transform has previously been discussed[3]. Consider the general function performed by A-O devices as defined by equation (1). Because the image $I(x,y)$ has been sampled by the sampling grid to produce $\hat{I}(x,y)$, the functions g and h must be shifted in frequency into the band occupied by the image. Hence, g and h cannot be base-band Walsh functions. The appropriate choice is

$$g(t) = \sin \omega_{o,x} t \; \mathrm{wal}(i,t) \tag{20}$$

$$h(t) = \sin \omega_{o,y} t \; \mathrm{wal}(j,t) \tag{21}$$

where $\omega_{o,x}$ and $\omega_{o,y}$ are the frequencies which will produce SAW's with wavelengths equal to the metal grid pattern in the x and y directions, respectively. These waveforms are shown in Figure 13.

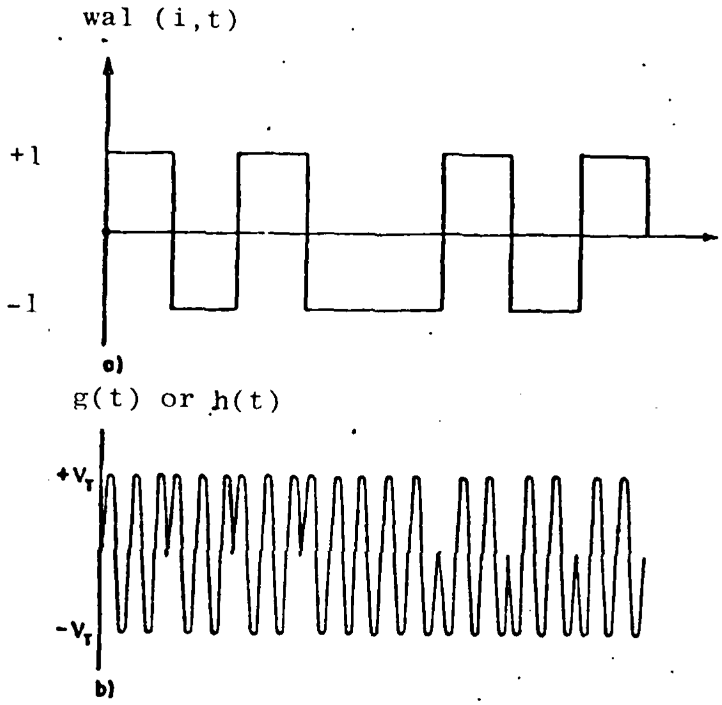


Figure 13 - Modulation of Sinusoid by a Walsh Function
a) Walsh function, b) Modulated Sinusoid

However, to be realistic, it is important to consider the limitations imposed by the bandwidth of the A-O device transducers. Since the Walsh functions are square waves they require unlimited bandwidth for exact representation. However, most of the spectral energy of these waveforms will lie below some maximum frequency. As long as the transducers pass frequencies within this bandwidth the A-O device can function as a Hadamard transformer. Walsh function spectrums have been derived in the form of recursive formulas[15]. In order to present these formulas some additional notation is required. First, define the cal and sal functions

$$\mathrm{cal}\ (n/2,t) = \mathrm{wal}\ (n,t);\ n \text{ even}. \tag{22}$$

$$\mathrm{sal}((n+1)/2,t) = \mathrm{wal}\ (n,t);\ n \text{ odd}. \tag{23}$$

Now define the Fourier transforms of these functions to be

$$C(n,f) = \int_{-1/2}^{+1/2} \mathrm{cal}\ (n,t)\ e^{-j\omega t}\ dt \tag{24}$$

$$S(n,f) = \int_{-1/2}^{1/2} \mathrm{sal}\ (n,t)\ e^{-j\omega t}\ dt \tag{25}$$

The interval $-1/2 \leq t \leq 1/2$ can be thought of as the time that the Walsh function acoustic wave will be on the active sensor area.

With these definitions the recursive formulas for C (n,f) and S (n,f) are given by

$$C(n,f) = \begin{cases} (-1)^{[n/1]} \cos(\pi f/2)\ C(n/2,f/2);\ n \text{ even} \\ j(-1)^{[n/2]} \sin(\pi f/2)\ S((n+1)/2,\ f/2);\ n \text{ odd} \end{cases} \tag{26}$$

$$S(n,f) = \begin{cases} -(-1)^{\left[\frac{n-1}{2}\right]} \cos(\pi f/2)\ S(n/2,f/2);\ n \text{ even} \\ -j(-1)^{\left[\frac{n-1}{2}\right]} \sin(\pi f/2)\ C((n-1)/2,f/2);\ n \text{ odd} \end{cases} \tag{27}$$

where $[n/2]$ means the largest integer equal to or smaller than n/2.

Using these formulas the Fourier transform of the Walsh functions can be easily computed. The first few of these are shown in Tables 1 and 2. Each of these functions is a product of sines and cosines multiplied by a term of the form $\sin^2(\pi f/k)/(\pi f/k)$. The bandwidth of the Walsh functions is

n	$j\,S(n,f)$
0	-
1	$\dfrac{\sin^2(\pi f/2)}{\pi f/2}$
2	$\dfrac{-\cos(\pi f/2)\ \sin^2(\pi f/4)}{\pi f/4}$
3	$\dfrac{-\sin(\pi f/2)\ \sin(\pi f/4)\ \sin^2(\pi f/8)}{\pi f/8}$
4	$\dfrac{-\cos(\pi f/2)\ \cos(\pi f/4)\ \sin^2(\pi f/8)}{\pi f/8}$
5	$\dfrac{-\sin(\pi f/2)\ \cos(\pi f/4)\ \sin(\pi f/8)\ \sin^2(\pi f/16)}{\pi f/16}$
6	$\dfrac{\cos(\pi f/2)\ \sin(\pi f/4)\ \sin(\pi f/8)\ \sin^2(\pi f/16)}{\pi f/16}$
7	$\dfrac{\sin(\pi f/2)\ \sin(\pi f/4)\ \cos(\pi f/8)\ \sin^2(\pi f/16)}{\pi f/16}$
8	$\dfrac{-\cos(\pi f/2)\ \cos(\pi f/4)\ \cos(\pi f/8)\ \sin^2(\pi f/16)}{\pi f/16}$

Table 1 - $S(n,f)$ vs. n

n	C(n, f)
0	$\frac{\sin(\pi f)}{\pi f}$
1	$\frac{\sin(\pi f/2)\ \sin^2(\pi f/4)}{\pi f/4}$
2	$\frac{-\cos(\pi f/2)\ \sin(\pi f/4)\ \sin^2(\pi f/8)}{\pi f/8}$
3	$\frac{\sin(\pi f/2)\ \cos(\pi f/4)\ \sin^2(\pi f/8)}{\pi f/8}$
4	$\frac{-\cos(\pi f/2)\ \cos(\pi f/4)\ \sin(\pi f/8)\ \sin^2(\pi f/16)}{\pi f/16}$
5	$\frac{-\sin(\pi f/2)\ \sin(\pi f/4)\ \sin(\pi f/8)\ \sin^2(\pi f/16)}{\pi f/16}$
6	$\frac{-\cos(\pi f/2)\ \sin(\pi f/2)\ \cos(\pi f/8)\ \sin^2(\pi f/16)}{\pi f/16}$
7	$\frac{-\sin(\pi f/2)\ \cos(\pi f/4)\ \cos(\pi f/8)\ \sin^2(\pi f/16)}{\pi f/16}$
8	$\frac{-\cos(\pi f/2)\ \cos(\pi f/4)\ \cos(\pi f/8)\ \sin(\pi f/16)\ \sin^2(\pi f/32)}{\pi f/32}$

Table 2 - C(n,f) vs. n

determined by this term. In general, sal transforms from $S(2^n+1,f)$ through $S(2^{n+1},f)$ contain a term of the form $\sin^2(\pi f/2^{n+2})/(\pi f/2^{n+2})$ while cal transforms from $C(2^n,f)$ through $C(2^{n+1}-1,f)$ contain a term of the form $\sin^2(\pi f/2^{n+2})/(\pi f/2^{n+2})$. The half power point of this envelope function occurs at

$$f_{1/2} = .5816 \times 2^{n+2} \tag{28}$$

(that is $\sin^2(\pi f_{1/2}/2^{n+2})/(\pi f_{1/2}/2^{n+2}) = 1/\sqrt{2}$.) If the number of Hadamard transform components along one axis is doubled then the corresponding transducer bandwidth must also be doubled. An identical statement can be made concerning bandwidth requirements for an A-O Fourier transformer.

Consider now the bandwidth required to excite Walsh functions on the A-O sensor. The sequency of $S(2^{n+1},f)$ and $C(2^{n+1},f)$ is 2^{n+1}. A sine or cosine wave defined on the same interval (i.e., $0 \leq x \leq 1$) also has a sequency (i.e., number of zero crossings). For example, a sine or cosine wave with sequency 2^{n+1} has frequency

$$f_s = .5 \times 2^{n+2} \tag{29}$$

From the discussion above, the frequency of this sinusoid is slightly lower than the half power point of $S(2^{n+1},f)$ and $C(2^{n+1},f)$. The maximum of the envelope occurs at $f = .3711 \times 2^{n+2}$ while the first zero of this function which is not at the origin occurs at $f = .6366 \times 2^{n+2}$. This is shown in Figure 14.

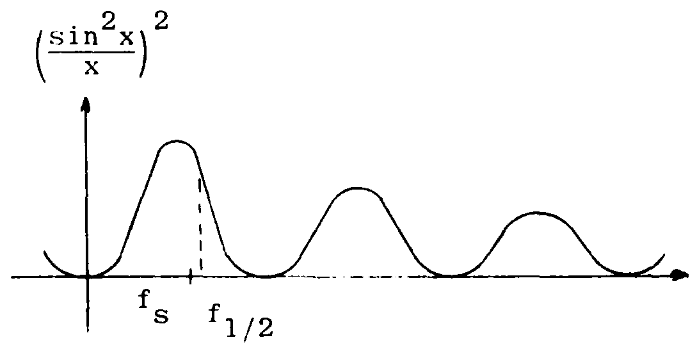


Figure 14 - Walsh/Fourier Bandwidth Relationship

However, because of the energy contained in the sidelobes of the envelope function it would seem advisable to require a bandwidth at least 2 or 3 times that required for sinusoids of the same sequency.

Because of the requirement of extra bandwidth, the use of A-O devices to compute Hadamard transforms is not as attractive as using these same devices to compute Fourier transforms. The utility of the Hadamard transform becomes important in implementations where it is costly to include multiplications. However, the A-O devices discussed in Section II are ideally suited for functional multiplication. Hence, using these devices to compute Hadamard transforms is probably not their best application.

2. Matched Filtering

The matched filter has long been used in communications systems as a means to filter a signal corrupted by additive noise. The extension of the matched filter to a two-dimensional signal (image) was first proposed by Vander Lugt[16,17]. Briefly, consider an image $I(x,y)$ consisting of signal s plus noise n.

$$I(x,y) = s(x,y) + n(x,y) \tag{30}$$

The matched filter is a linear filter with impulse response $h(x,y)$ which filters the image in such a way to maximize the signal-to-noise ratio at the output. Let $F(\omega_x,\omega_y)$, $S(\omega_x,\omega_y)$, $N(\omega_x,\omega_y)$ and $H(\omega_x,\omega_y)$ be the two-dimensional Fourier transforms of I, s, n and h respectively. The noise is usually characterized by its power spectrum NN^*. The signal-to-noise ratio is defined to be

$$\text{signal-to-noise} = \frac{\left|\iint SHe^{j(\omega_x x+\omega_y y)}\, d\omega_x d\omega_y\right|^2}{\iint NN^*\, HH^*\, d\omega_x d\omega_y} \tag{31}$$

The signal-to-noise ratio is maximized over the filter output space if

$$H(\omega_x,\omega_y) = \frac{S^*(\omega_x,\omega_y)}{NN^*} \tag{32}$$

the filter output is

$$m(x,y) = \iint FHe^{j(\omega_x x+\omega_y y)} \, d\omega_x d\omega_y \qquad (33)$$

If the noise is white, (NN*, a constant), then the matched filter output is simply the correlation between the image and the desired object to be detected.

Now if the image contains the signal but offset to position $\Delta x, \Delta y$ then the correlation in the output will be shifted to $-\Delta x$, $-\Delta y$. There will be a peak in the output at this position. If the image contains multiple signals at various positions within the image then the output will also contain a number of correlation peaks. The matched filter is translation invariant in a limited sense. If the image contains a signal (feature) then the output will contain a correlation peak. However, if the signal is translated then the output peak is also translated. Strictly speaking, the output would be different (i.e., translated) so that the matched filter is not translation invariant. However, it is invariant in the sense that the correlation peak will occur <u>somewhere</u> only if the image contains the signal. Another way of looking at it is to say that the matched filter preserves the positional information about the signal. There is no dimensionality reduction since the matched filter output is a function of x and y. However, signal detection is improved.

Referring back to Section III A, if the image is scanned and broken into a number of views it may not be required to locate the position of a feature within a view. Rather,, the <u>detection</u> of a feature somewhere in the view may be adequate. Hence, the matched filter may preserve too much information. Perhaps by discarding this information a simpler implementation may result.

Now consider implementation of the matched filter using A-O devices. When the two-dimensional matched filter was first discussed by Vander Lugt he proposed implementing it using coherent optics. In this implementation the image is

first transformed using a Fourier transformer lens. The matched filter is then implemented in the Fourier domain by a spatial filter. The filtered transform is then passed through a second Fourier transformer lens. The output image contains bright spots corresponding to signals in the input image. This is a rather natural implementation for an optical processor since the various lenses and filters can be lined up on an optical bench and the information passes through the system in a parallel fashion. That is, the data stream is a two-dimensional light pattern throughout the processor.

Because A-O devices can compute two-dimensional Fourier transforms, an analogous system could be built using these devices. A block diagram of such a system is shown in Figure 15. The image is viewed by an A-O device such as a Deft sensor. The sensor is used to compute the two-dimensional Fourier transform of the image. The transform is raster scanned by properly addressing A-O sensor No. 1. This sensor will have two outputs which are the real and the imaginary part of the Fourier transform. These transform components are multiplied by the corresponding components of the matched filter which are stored in a memory. A complex multiplication operation consisting of four real multiplies is required. The weighted transform components are then input to two CRT's. Since the transforms are raster scanned, the data stream is in the correct format for the CRT. The real and imaginary parts of the Fourier transform, weighted by the matched filter, are displayed on CRT No. 1 and No. 2 respectively. These functions are then viewed by A-O devices No. 2 and No. 3. It may be possible to directly attach these sensors to fiber optic faceplate CRT's. Since the functions on the CRT's must remain constant for the period of time required to scan out the inverse transforms from the A-O devices, storage CRT's are required. The output of A-O devices Nos. 2 and 3 are then A/O converted and input to the decision processor. The decision processor searches for correlation peaks in the

matched filter output and compares these peaks with a threshold valve in order to determine if the feature is contained in the image.

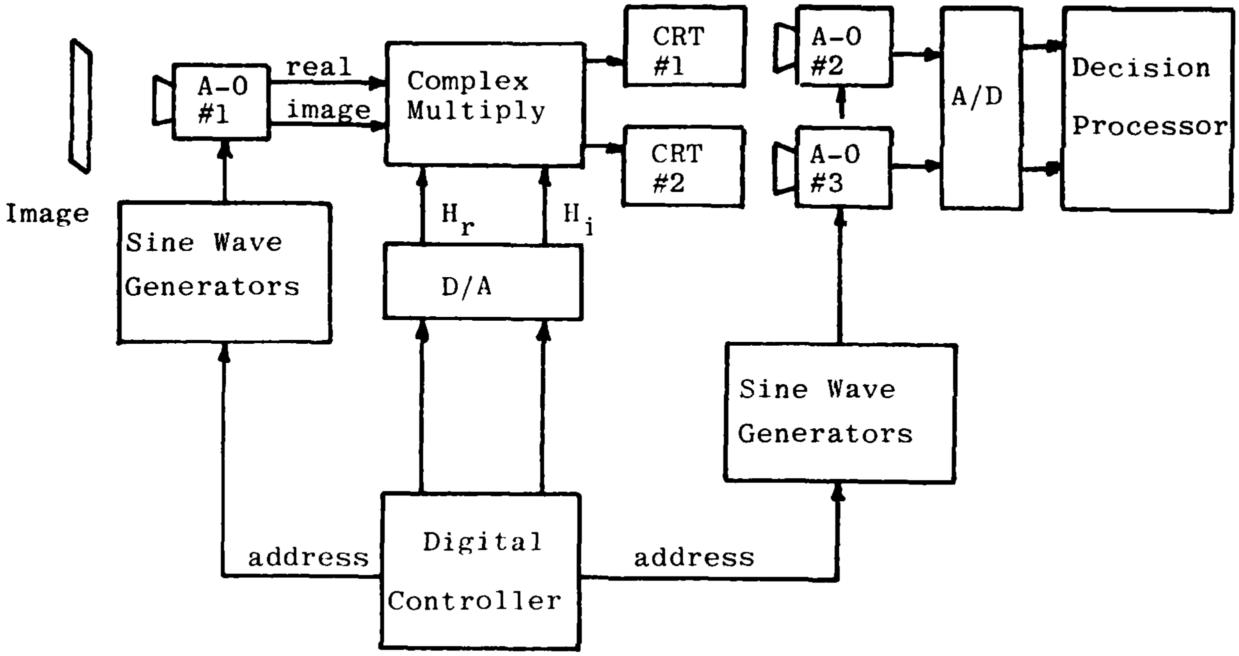


Figure 15 - Matched Filter: A-0 Implementation

A disadvantage of this implementation is the requirement of three A-0 devices and the need to generate a pictorial representation of the weighted image transform in order to utilize the second two A-0 devices.

In order to avoid this complexity, consider the alternative of computing the matched filter in the image domain rather than the transform domain. The matched filter output is the convolution of I(x,y) with h(x,y). That is,

$$m(x,y) = \int\int I(\alpha,\beta)h(\alpha-x,\beta-y)d\alpha d\beta \qquad (34)$$

In order to compute this function by use of an A-O device which implements equation (1) the function h(x,y) must be decomposable into

$$h(x,y) = g(x)h(y) \tag{35}$$

However, this will not generally be the case for arbitrary features to be detected. Hence, this approach cannot be taken.

In order to avoid the requirement for an inverse transform a third approach can be taken. As has been pointed out, the matched filter output preserves positional information which may not be required. Perhaps some implementation advantage can be gained by dispensing with this information. To this end, consider the matched filter for the autocorrelation function of I(x,y). The transform of this function is given by

$$FF^* = (S+N)(S+N)^* = SS^*+SN^*+S^*N+NN^* \tag{36}$$

If the signal and the noise can be considered uncorrelated then $SN^* = S^*N = 0$ and

$$FF^* = SS^* + NN^* \tag{37}$$

In this case the signal is the autocorrelation function and the noise is NN^*. The function FF^* consists of signal plus added noise. Hence, the matched filter can be used to filter the signal from the noise. In this case the matched filter is given by

$$\hat{H}(\omega_x,\omega_y) = \frac{(SS^*)^*}{(NN^*)(NN^*)^*} = \frac{(SS^*)^*}{(NN^*)^2} \tag{38}$$

The filter output is

$$\hat{m}(x,y) = \iint FF^*\hat{H}e^{j(\omega_x x+\omega_y y)}\, d\omega x d\omega y \tag{39}$$

However since F has been replaced by FF^*, any correlation peak indicating the signal or feature with transform SS^* will occur at x = y = 0 at the matched filter output. Hence, it is only necessary to compute

$$\hat{m}(o,o) = \iint FF^*\hat{H}d\omega_x d\omega_y \tag{40}$$

This function is simpler to implement using A-O devices. A functional block diagram is shown in Figure 16. The image is viewed by an A-O device. The device again is used to compute the two-dimensional Fourier transform of the image. The sensor electronics are designed to output the magnitude of the Fourier transform rather than the real and imaginary parts. Computing the magnitude actually required less electronics than computing the real and imaginary part. This is because, in the second case, phase information must be preserved and a synchronous detector is required. Such a detector for a Deft sensor is described in reference 18, Section II. In the matched filter application the A-O device is scanned over the range of spatial frequencies of interest. The sensor output is squared and multiplied by the corresponding values of the matched filter $\hat{H}$. Both the sensor output and $\hat{H}$ values are real which simplifies this computation. The output of the second multiplier is then integrated over all spatial frequencies. The integrator output is compared with a threshold value to determine if the signal (feature) is present.

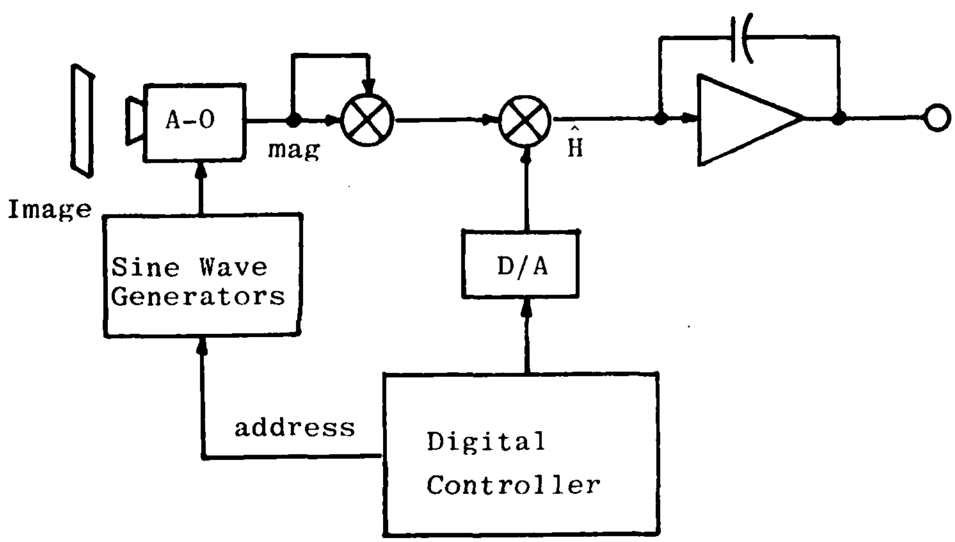


Figure 16 - FF* Matched Filter : A-O Implementation

The simplification in implementation when F is replaced by FF* is obvious from a comparison of Figure 15 with Figure 16. In the second case no CRT's are required. In addition, one rather than three A-O devices are required. The matched filter for SS* has some functional advantages and disadvantages when compared with the matched filter for S. The advantages are the following. First, there is a dimensionality reduction from input to output. This simplifies the decision processor. (A two-dimensional search is not required.) Second, the matched filter for SS* is invariant to translation which also simplifies the decision processor. The disadvantage of this matched filter is that the detection of a signal with transform SS* does not imply that the image contained the signal with transform S. All signals with transforms $|S|e^{j\phi(\omega_x,\omega_y)}$ will produce the same result at the matched filter output. ϕ is an arbitrary phase function. Hence, this matched filter can be used to correctly detect features with transform S but will produce false alarms when certain other images are viewed. This disadvantage may be acceptable if the purpose of the feature extractor is to screen photos with a final decision on feature content being made by another means such as human observations.

As a final comment, neither types of matched filters considered here are invariant to feature rotation or scale change.

3. Method of Invariant Fourier Signatures

In this section some prefilter algorithms will be developed which compute feature signatures which are invariant to feature translation, rotation and scale. These algorithms are refinements of the matched filter of the image autocorrelation function which was developed in the previous section. Referring to equation (38), $\hat{H}$ can be thought of as a weight function which is directly proportional to SS* and inversely proportional to $(NN*)^2$. The matched filter output at $\hat{m}(o,o)$ is simply the integral over all spatial frequencies of the

product of this weight function with FF*. As a result, all of the information contained in the fine structure of FF* is lost and not available to the decision processor. In fact, all the decision processor gets is a single number on which the decision is to be based.

There are some obvious drawbacks to this type of preprocessor. First, this matched filter is rather insensitive to the shape of the transform FF*. That is, there may be two dissimilar images I_1 and I_2 with transforms F_1 and F_2 such that

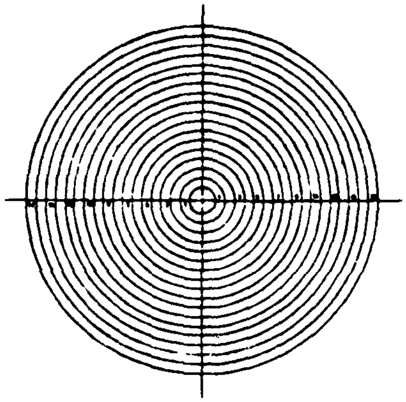
$$F_1F_1^* \neq F_2F_2^* \tag{41}$$

and yet

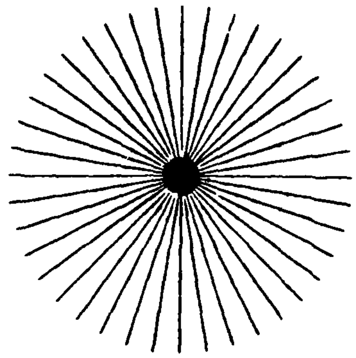
$$\iint F_1F_1^*\hat{H}d\omega_x d\omega_y = \iint F_2F_2^*\hat{H}d\omega_x d\omega_y \tag{42}$$

This would lead to additional, undesirable false alarms. A second drawback which has been mentioned is that the matched filter is not invariant to either rotation or scale changes. Hence, to use matched filters for feature extraction a number of filters must be implemented for different angular orientations and scale factors. Matched filter degradation due to rotation and scale mismatch is discussed in reference 19.

Poor sensitivity of the preprocessor transform shape results because the two-dimensional image is reduced to a scaler. Better sensitivity may be obtainable if the preprocessor is modified so that its output is a vector rather than a scaler. A promising approach first taken by Lendaris and Stanley[20] is to replace the double integral in equation (40) with n line integrals. That is, the product $FF^*\hat{H}$ is integrated along n contours. These integrals are then the components of a n-vector which is the prefilter output. Two of the contours used by Lendaris and Stanley are shown in Figure 17.



(a)
Concentric Circles



(b)
Radial Lines

Figure 17 - Integration Contours

In their work they only used the magnitude of the Fourier transform. They did not implement the matched filter $\hat{H}$. They were concerned with computing prefilter outputs which are invariant to translation and rotation or to translation and scale but not both. In this section, a more general type of prefilter will be developed which can be invariant to translation, rotation and scale and also allow noise filtering which is analogous to matched filtering. However, to clarify the presentation some algorithms of Lendaris and Stanley will first be developed. These will then be generalized.

To begin, assume that either $|F|$ or FF^* is available from an A-O sensor. Also assume for now that $F = S$. That is, there is no noise and the image consists of only the feature to be detected. Now if the feature is translated by $\Delta x, \Delta y$ then the transform changes to $Fe^{-j(\Delta_x \omega_x + \Delta_y \omega_y)}$. Hence both $|F|$ and FF^* do not change and these functions are said to be invariant to translation. We will now consider the two sets of contours of Figure 17 separately.

Consider first the set of concentric circles. Let the i-th circle have radius r_i. The difference $r_i - r_{i-1}$ need not be a constant. There are n contours. Let FF^* or $|F|$ be integrated around any of these contours. Now if the feature were to rotate through angle θ then this integral would not change.

This is because FF* or $|F|$ would only rotate around the transform origin by the same angle. This will be true for all contours. Hence, the n-vector whose components are the n line integrals is invariant to rotation.

Now consider the second set of contours. Let the i-th radial line be at an angle θ_i from the ω_x-axis. The difference $\theta_i - \theta_{i-1}$ need not be a constant. There are n contours. Let FF* or $|F|$ be integrated along any of these contours. Now if the feature were to be scaled in size by α then $|F|$ and FF* would be scaled in size by $1/\alpha$. As long as the radial lines extended far enough from the origin so that all the significant energy of the transform will be included in the contour integration for both scaled and unscaled feature, then in both these cases the line integration will remain approximately the same. This will be true for all contours. Hence, the n-vector who's components are the n line integrals is (approximately) invariant to scale.

Prefilters based on these algorithms can be implemented easily using A-O sensors. The sensor spatial frequency address would be incremented along a contour. The sensor output would then be integrated along the contour. A simple integration formula such as the trapezoidal rule

$$\frac{1}{h_s}\left[\sum_i f(x_i) - \frac{1}{2}(f(a)+f(b))\right] \approx \int_a^b f(x)dx \tag{43}$$

where $h_s = x_i - x_{i-1}$ would probably be adequate. (Note that the Thomson - CSF sensor would be limited to scanning along radial lines. For a discussion refer to Section II D.)

The invariant properties of these algorithms appear to be very useful. They are also noteworthy because only a single Fourier transform is required and, therefore, they can be naturally implemented using a single A-O device. These algorithms were successfully used by Lendaris and Stanley to detect features in aerial photographs. They have also been used by Pernick, et al, in the screening of cervical cytological

samples[21]. However, these algorithms have some disadvantages.

The first disadvantage is that they do not consider the effect of noise. In the matched filter the output is enhanced by multiplying FF* by the weight function $\hat{H}$. If the image contains signal plus noise then it is not possible to weight FF* by $\hat{H}$ before integrating along the contours because the feature is assumed to be either rotated or scaled by some unknown factor. Since $\hat{H}$ is not invariant to these changes in the feature, the required weighting function is not known. The proper place to apply filtering is after the line integrations. Assume that signal and noise are uncorrelated so that equation (37) is satisfied. Since the integral of a sum equals the sum of the integrals then signal and noise will still be additive at the output of the contour integrations. (This implies that FF* is used. If $|F|$ is used, signal and noise are no longer additive.) That is,

$$\phi_i FF^* = \phi_i SS^* + \phi_i NN^* \tag{44}$$

where ϕ_i is the i-th contour integral. Hence, the matched filter principle can be applied at the prefilter output after line integration. Let the weight to be applied to the output of the i-th line integral be w_i. Then from equation (32)

$$w_i = \frac{\phi_i SS^*}{(\phi_i NN^*)^2} \tag{45}$$

To be specific, consider the case of concentric circles for contours. The feature to be detected has transform S. Since the algorithm has been shown to be invariant to feature rotation, w_i will also be invariant to feature rotation. (The noise does not change with feature rotation since feature and noise are assumed uncorrelated.) Hence, the set of weights is a constant n-vector which only depends on the noise and on the feature but not on the angular orientation of the feature. That is, the weight vector can be computed apriori. This will also be

the case when the contours consist of radial lines. In this case the feature has an arbitrary scale. However, the algorithm is invariant to scale so that $\phi_i SS^*$ is constant. Hence the weight vector is again constant and can be computed apriori using equation (45)

In general, let the i-th component of the measurement vector be m_i where

$$m_i = \phi_i FF^* \tag{46}$$

then the prefilter output n-vector is given by V where

$$V^T = |w_1 m_1, w_2 m_2, ---, w_n m_n| \tag{47}$$

Given that matched filtering can be applied to these algorithms they still suffer a disadvantage in the case that the feature may have arbitrary position, rotation and scale. These algorithms are invariant to translation and rotation or translation and scale but not to all three. If the feature is present but suffers all three changes with respect to the reference feature with spectrum S then the output n-vector will not correspond to the reference output and detection may not be possible. It is, however, possible to generalize these algorithms so that the prefilter can be made invariant to all three feature changes.

To develop the more general algorithm, begin with the image I(x,y) with transform $F(\omega_x, \omega_y)$. Form either $|F|$ or FF^* which has been shown to be invariant to translation. (If matched filtering is to be applied later then FF^* should be used.) Then form the function

$$G(\rho,\theta) = |F(e^\rho \cos\theta,\ e^\rho \sin\theta)| \tag{48}$$

or if FF* was formed,

$$G(\rho,\theta) = F(e^\rho \cos\theta,\ e^\rho \sin\theta)\ F^*(e^\rho \cos\theta,\ e^\rho \sin\theta) \tag{49}$$

That is, G is $|F|$ or FF^* distorted exponentially in radius and expressed in polar coordinates. The function G is periodic in θ with period π. Assume now that the image consists of signal

with no noise. Consider now the change in G when the signal or feature in $I(x,y)$ suffers an arbitrary rotation and scale change. Let (r,γ) be the location of an arbitrary component of $|F|$ or FF^* before rotation and scale change. The location of the corresponding component in G is $(\ln r,\gamma)$. After rotation of the feature through angle ϕ and scaling by α the component in $|F|$ or FF^* will move to $(r/\alpha,\gamma+\phi)$. The location of the corresponding component in G is $(\ln r - \ln\alpha,\ (\gamma+\phi)_{\mathrm{mod}\,\pi})$. That is, G will be translated by $-\ln\alpha$ along the ρ direction and by ϕ along the θ direction (modulo π). Since $|F|$ or FF^* is invariant to translation, any combination of translation, rotation and scale change of the feature will result in only a translation in G. (G will also suffer a gain change to $|\alpha|^{-2}G$ but this is not important.)

Since the only change in G is a translation, by taking a second two-dimensional Fourier transform, this time of $G(\rho,\theta)$, and then forming the magnitude or magnitude squared of this second transform, a function is formed which is invariant to translation in G. Hence, this last function is invariant to translation, rotation and scale of the feature. This function is

$$H(\omega_\rho,\omega_\theta)=\int_{\rho_{min}}^{\rho_{max}}\int_{-\pi/2}^{\pi/2} G(\rho,\theta)e^{-j(\omega_\rho\rho+\omega_\theta\theta)}d\rho\, d\theta \tag{50}$$

The functions $|H|$ and HH^* are invariant to translation of G. Strictly speaking, this is true only along the lines parallel to the ω_ρ-axis defined by

$$\omega_\theta = 2n;\ n \text{ an integer} \tag{51}$$

This is because G is periodic in θ and should be expanded in a Fourier series rather than a Fourier integral in the θ-direction. The Fourier integral evaluated on the above lines reduces to the Fourier series.

Although $|H(\omega_\rho,\ n/\pi)|$ or $H(\omega_\rho,n/\pi)H^*(\omega_\rho,n/\pi)$ is a two-dimensional invariant function it is probably not necessary (and certainly not desirable from a computational standpoint) to compute this function. To develop more easily computable

invariant signatures, recourse is made to the Fourier transform projection theorem. Let $P_\phi[G(\rho,\theta)]$ be the projection of G onto a line at angle ϕ from the ρ-axis. The projection theorem states that the one-dimensional Fourier transform of $P_\phi[G]$ equals $H(\omega_\rho,\omega_\theta)$ restricted to a line through the origin of $(\omega_\rho,\omega_\theta)$ - space and at an angle ϕ to the $\omega\rho$-axis[22]. This line is shown dashed in Figure 18. Since $G(\rho,\theta)$ is periodic in θ, $P_\phi[G]$ will also be periodic for $\phi \neq 0$. Hence, $P_\phi[G]$ should be expanded in a Fourier series rather than a Fourier transform. Let $F\{P_\phi[G]\}$ be the Fourier series expansion of $P_\phi[G]$. The terms of this series equal $H(\omega_\rho,\omega_\theta)$ evaluated at the intersection of the dashed line with the horizontal lines $\omega_\theta=2n$.

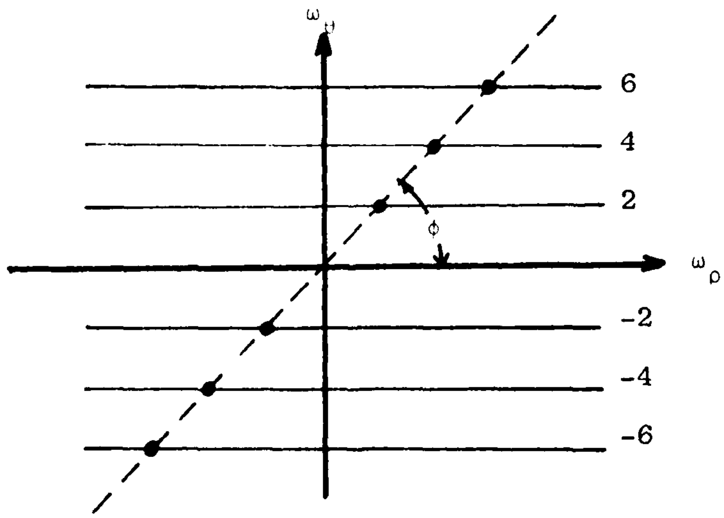


Figure 18 - $(\omega_\rho,\omega_\theta)$ - Space and $F\{P_\phi\}$ Line

Since $|H|$ and HH^* are invariant, so is $|F\{P_\phi\}|$ and $F\{P_\phi\}F^*\{P_\phi\}$. Now $F\{P_\phi\}$ contains only some of the information contained in H. It is conjectured that by properly choosing ϕ values experimentally, a set of invariant signatures could be developed which contain most of the useful information in H. The signature or signatures to be used may depend on the set of features to be recognized.

Because of the way this development has been carried out, it may not be clear how the invariant signatures are to be computed from the first Fourier transform $F(\omega_x, \omega_y)$. This can be clarified with the aid of Figures 19 and 20. In the following it will be assumed that all processing after computation of the Fourier transform of the image will be done digitally. Hence, continuous functions must be sampled to get discrete samples for digital processing.

Figure 19 shows the domain of the pertinent part of F in (ω_x, ω_y) - space. F has been restricted to spatial frequencies ω_x, ω_y where $\rho min \leq \sqrt{\omega_x^2+\omega_y^2} \leq \rho max$. Because of the mapping $\rho = e^{\hat{\rho}}$, F will be only sampled on the concentric circles shown in Figure 19. There are n circles and their radii are e^k, e^{2k}, e^{3k}, ..., e^{nk} where

$$k = \ln\rho_{min} \tag{52}$$

$$\rho_{max} = e^{nk} \tag{53}$$

Because of the mapping from F to G, these circles in (ω_x, ω_y) -space will map to the dashed, equally spaced vertical lines shown in Figure 20 in (ρ,θ) - space. Since the image is a real intensity function

$$F(\omega_x, \omega_y) = F(-\omega_x, -\omega_y) \tag{54}$$

so that G is periodic in the θ-direction with period π.

Consider now the formation of the projection $P_\phi[G]$. P_ϕ is a function of a single variable μ where

$$\mu = \rho cosec\ \theta \tag{55}$$

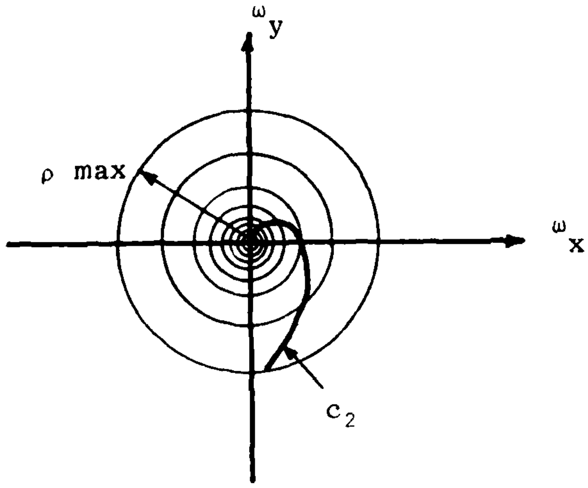


Figure 19 - (ω_x, ω_y) - Space Contours

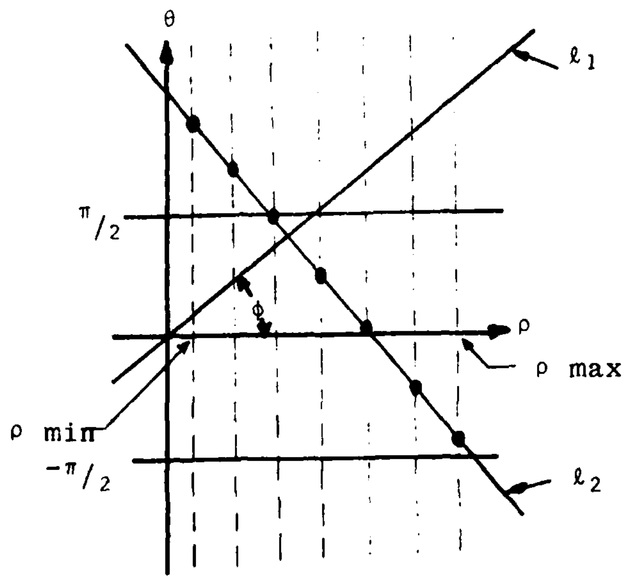


Figure 20 - (ρ, θ) - Space Contours

The domain of P_ϕ is the line through the origin of (ρ,θ) - space at angle θ to the ρ-axis. This line is labeled ℓ_1 in Figure 20. P_ϕ will only be computed at a discrete set of points. Let μ_i be one of these points. To determine $P_\phi(\mu_i)$ a straight line is drawn which passes through the point μ_i and is at right angles to ℓ_1. This line is labeled ℓ_2 in Figure 20. Now, to find $P_\phi(\mu_i)$, simply integrate G along ℓ_2. Since G is only available along the vertical, dashed lines of Figure 20, a numerical integration is performed using sample points which are the intersection of ℓ_2 with the vertical dashed lines. The values of G at this set of points is simply the function $|F|$ or FF* evaluated at a set of points which is defined by the intersection of the n, concentric circles in Figure 19 with the curve which is the mapping of ℓ_2 from (ρ,θ) - space to (ω_x,ω_y) - space. This curve is labeled c_2 in Figure 19. Note that c_2 is a spiral. This is always the case unless $\phi = 0^o$ or $\phi = 90^o$. If $\phi = 0^0$ then ℓ_2 is parallel to the vertical, dashed lines in Figure 20. In that case μ_i is chosen so that ℓ_2 coincides with one of these vertical lines. That is, c_2 will be a concentric circle. This implies the $|F|$ or FF* be integrated around a circle in order to compute $P_\phi(\mu_i)$. If $\phi = 90^o$ then ℓ_2 is parallel to the ρ-axis. In (ω_x,ω_y) - space the corresponding c_2 will be a radial line. The intersection of this line with the n, concentric circles defines the sample points used to compute the numerical integration. These special cases where $\phi = 0^o$ or 90^o result in contours which are similiar to the contours of Figure 17. The only difference is the exponentially distorted spacing of circles. For any other value of $0^o < \phi < 90^o$ the integration contours are spirals. Hence, this algorithm is more general than the previously considered one.

Now for whatever ϕ angle used, the above procedure is repeated for each of the sample points μ_i. In this way, $P_\phi[G]$ is computed at a set of sample values. To increase efficiency

of the remaining computation, the number of sample values of P_ϕ is chosen to be a power of two. Now recall that the one-dimensional Fourier series of $P_\phi[G]$ equals $H(\omega_\rho, \omega_\theta)$ restricted to a line through the origin of $(\omega_\rho, \omega_\theta)$-space and at an angle ϕ to the ω_ρ-axis. Hence, the final step in the algorithm is to compute the Fourier series of P_ϕ. Since P_ϕ is only computed at a set of equally spaced samples, μ_i, a discrete Fourier transform (DFT) rather than a continuous Fourier series is computed. As long as the sample spacing is sufficiently fine to prevent significant aliasing, the substitution of the discrete Fourier transform for the Fourier series is acceptable. Since P_ϕ has been sampled at 2^m points, the DFT can be computed using an FFT algorithm.

In the above development, the functions F, G and H have been defined. However, in the actual calculations of the invariant signature $F\{P_\phi[G]\}$ only the function F need be considered. In review, to compute the signature the following steps are required:

1) A set of radii defined by

$$r_i = e^{ik} \tag{56}$$

for some constant k is chosen. These radii define the n, concentric circles. (For the special case that $\phi = 0^\circ$, n is chosen to be a power of two.)

2) A projection angle $0^\circ \leq \phi \leq 90^\circ$ is chosen.

3) An integer m is chosen to define the FFT length as 2^m.

4) If $\phi < 90^\circ$ then the following step is performed: For each $0 \leq i \leq 2^m - 1$ $|F|$ or FF^* is integrated along the contour c_2 using samples with (polar) coordinates

$$(e^{\ell k}, (\ell-1)\phi + i\pi/2^m) \quad 1 \leq \ell \leq n \tag{57}$$

The result is a vector of length 2^m with each element of the vector corresponding to a difference value of i.

5) If, instead, $\phi = 90^\circ$ then the following step is performed: For each $0 \leq i \leq n$ $|F|$ or FF^* is integrated around each of the n,

concentric circles. A discrete integration is performed by only sampling F discretely around each circle. n is chosen to be 2^m so that the resulting vector is of length 2^m. Each element corresponds to a different value of i.

6) An FFT is performed on the 2^m -vector.

7) Either the magnitude or the magnitude squared of the FFT output samples are computed.

The resulting 2^m -vector of real numbers is the desired signature. The above series of steps can be repeated for other values of ϕ, if so desired. The above algorithm will be termed <u>the method of invariant Fourier signatures</u> (IFS). Elements of this algorithm are developed in a series of papers [23, 24, 25, 26, 27]. The procedure of exponentially distorting a function and then computing its Fourier transform can be shown to be equivalent to the Mellin transform[23].

$$MT(\mu) = \int_0^\infty f(x)x^{-j\mu-1}\,dx \tag{58}$$

In the above development the question of noise corruption was not considered. The invariant signatures which can be computed from the algorithm are only invariant in the noise-free case. We have already considered the application of the matched filter to the output of the prefilter defined by equation (47). Consider now the extension of this development to the more general prefilters developed above.

To begin, assume that the image I(x,y) consists of signal plus noise. That is, I is defined by equation (30). Assume also that signal and noise are uncorrelated so that equation (37) holds. Now let $\oint_i FF^*$ be the result of integrating FF^* along the i-th contour which may be a radial line, circle or spiral. Then with the above assumptions equation (44) still holds. That is, the exponential distortion of the radius does not effect this result. Define signal and noise to be $\tilde{s}(\mu)$ and $\tilde{n}(\mu)$ where

$$\tilde{s}(\mu_i) = \oint_i SS^* - \frac{1}{n}\sum_{k=1}^{n} \oint_k SS^* \tag{59}$$

$$\tilde{n}(\mu_i) = \oint_i NN^* - \frac{1}{n} \sum_{k=1}^{n} \oint_k NN^* \qquad (60)$$

That is $\tilde{s}(\mu_i)$ equals the result of the line integral at μ_i with the mean value over all i removed. The same applies to $\tilde{n}(\mu_i)$. (Removal of the mean has not been considered to this point.) Define the transforms of $\tilde{s}$ and $\tilde{n}$ to be $\tilde{S}$ and $\tilde{N}$ respectively. That is

$$\tilde{S} = F\{\tilde{s}\} \qquad (61)$$

$$\tilde{N} = F\{\tilde{n}\} \qquad (62)$$

and define

$$\tilde{F}\{P_\phi[G]\} = \tilde{S}+\tilde{N} \qquad (63)$$

($\tilde{F}\{P_\phi\}$ is simply $F\{P_\phi\}$ with the zero frequency term set to zero.) Now matched filtering can be applied since signal and noise are additive which follows from equation (44). However, to avoid another transform, we make the further assumption that $\tilde{S}$ and $\tilde{N}$ are uncorrelated so that

$$\tilde{F}\{P_\phi\}\tilde{F}^*\{P_\phi\} = \tilde{S}\tilde{S}^* + \tilde{N}\tilde{N}^* \qquad (64)$$

The removal of the means was necessary to make this assumption a possibility. The corresponding matched filter weight function is

$$\tilde{H} = \frac{\tilde{S}\tilde{S}^*}{(\tilde{N}\tilde{N}^*)^2} \qquad (65)$$

The matched filter output is

$$\tilde{m}(o) = \int \tilde{F}\{P_\phi\}\ \tilde{F}^*\{P_\phi\}\tilde{H}dv \qquad (66)$$

where v is the frequency variable.

Notice that the result is a scaler rather than a 2^m vector. What has happened is that by adding matched filtering, the decision processor algorithm has been included with the preprocessor algorithm. The only additional computation required is to compare $\tilde{m}(o)$ with a threshold. Note the form of $\tilde{H}$. It is the output which would occur if only the signal (feature) were present divided by the square of the output due to the

noise alone. Hence, where a large noise contribution was expected the output would be de-emphasized. $\hat{m}(o)$ is simply the inner product of the prefilter output with the matched filter weight.

Finally, consider the implementation of any of these algorithms using A-O devices. A block diagram is shown in Figure 21.

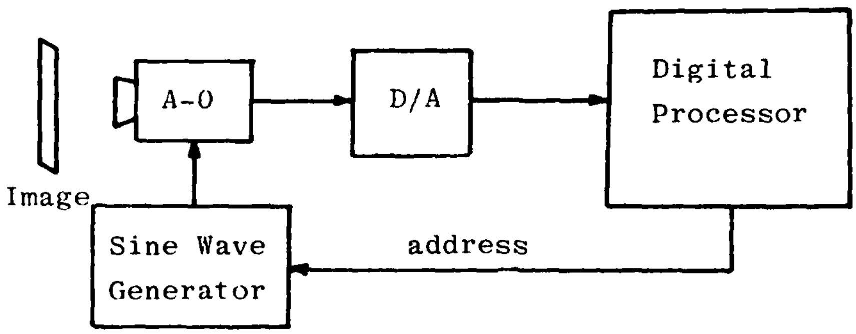


Figure 21 - Invariant Fourier Signatures: Implementation

The block diagram is rather simple. The A-O device is used to produce components of the two-dimensional Fourier transform of the image. The sample points are controlled by addressing the sensor. The remainder of the processing takes place in the digital processor. The amount of data memory required is 2^{m+1} words to hold the complex 2^m vector. Spatial frequency addresses can either be computed using a cordic algorithm[28] or precomputed and stored in ROM. It can be seen that these algorithms are well-suited for implementation with A-O devices since the two-dimensional image information is immediately reduced to a much smaller set of numbers which can be post-processed in the digital processor.

In summary, a flexible class of algorithms has been developed which can be used as a feature extractor prefilter. These algorithms can be translation, rotation and scale

invariant. This is strictly true only in the case where no noise is present. However, matched filtering can be used to minimize the noise at the prefilter output. The parameter ϕ can be used to optimize the prefilter for the particular feature or class of features to be identified in the image set. The parameter ϕ can be arbitrarily chosen because A-O devices can be addressed at arbitrary spatial frequencies. Finally, these algorithms are well-suited for implementation using an A-O device for a Fourier transformer and a mini-computer as a post-processor.

4. Method of Invariant Moment Signatures

The method of moments is used in a number of disciplines. The utility of moments in feature extraction was first pointed out by Hu[12]. It can be shown that algebraic combinations of image moments are invariant to translation, rotation and scale. This section is a discussion of image moments and how they can be computed using A-O devices.

The two-dimensional (p+q)-th moments of an image intensity function I(x,y) are defined by

$$m_{p,q} = \iint x^p y^q \; I(x,y) \; dxdy \tag{67}$$

It can be shown that the double sequence of moments $\{m_{p,q}\}$ is uniquely determined by I(x,y) and conversely, I(x,y) is uniquely determined by $\{m_{p,q}\}$ as long as I(x,y) satisfies certain conditions which are always met with real images[12]. A method of reconstructing I(x,y) from $\{m_{p,q}\}$ is given in reference 13.

The purpose of computing image moments is two-fold. First, the invariant functions of moments are desirable for feature extraction. Second, the possibility exists of replacing the image, which contains a large amount of information represented by its pixel values, with a much smaller set of moments $\{m_{p,q}$ such that $p \leq p_{max}$ and $q \leq q_{max}\}$. This is analogous to computing the two-dimensional Fourier

transform for the purpose of reducing the image information to a much smaller set of spatial frequencies.

Consider now the central moments which are defined by

$$\mu_{p,q} = \int\int (x-\bar{x})^p (y-\bar{y})^q \ I(x,y)d(x-\bar{x})d(y-\bar{y}) \tag{68}$$

where

$$\bar{x} = m_{1,0}/m_{0,0} \tag{69}$$

$$\bar{y} = m_{0,1}/m_{0,0} \tag{70}$$

It is well known that the central moments are invariant under translation of coordinates. Hence, the central moments are translation invariant. The central moments can be expressed in terms of the ordinary moments. For example

$$\mu_{0,0} = m_{0,0} \tag{71}$$

$$\mu_{1,0} = \mu_{0,1} = 0 \tag{72}$$

$$\mu_{2,0} = m_{2,0} - m_{0,0} \ \bar{x}^2 \tag{73}$$

$$\mu_{0,2} = m_{0,2} - m_{0,0} \ \bar{y}^2 \tag{74}$$

$$\mu_{1,1} = m_{1,1} - m_{0,0} \ \bar{x}\bar{y} \tag{75}$$

Similar expressions are easily derived for all higher order moments with the use of equation (68). If, for example, the ordinary moments could be computed using an A-O device then the central moments could be formed in a digital post processor.

Consider the information contained in the first few moments $\{\mu_{0,0}, \mu_{1,0}, \mu_{0,1}, \mu_{2,0}, \mu_{0,2}, \mu_{1,1}\}$

$$\mu_{0,0} = m_{0,0} = \int\int I(x,y)dxdy \tag{76}$$

represents the total image power. Both $\mu_{1,0}$ and $\mu_{0,1}$ are zero. However, the ordinary moments

$$m_{1,0} = \int\int xI(x,y)dxdy \tag{77}$$

$$m_{0,1} = \int\int yI(x,y)dxdy \tag{78}$$

locate the image centroid which is $(\bar{x},\bar{y})$ where $\bar{x}$ and $\bar{y}$ have already been defined.

The above set of moments characterize the size, gross shape and orientation of the image. If only these moments are considered then the image moments are identical to the moments of an image consisting of the ellipse shown in Figure 22[13]. The parameters of this ellipse are given by

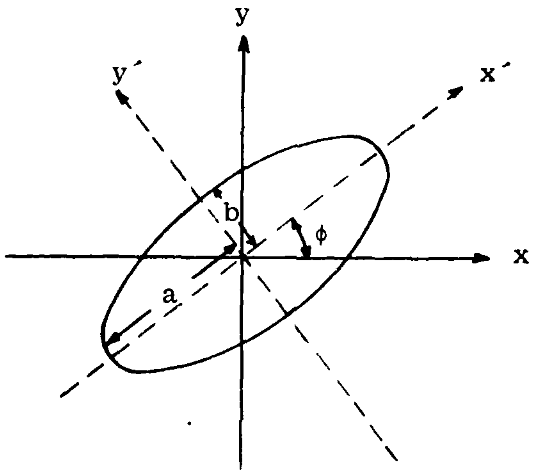


Figure 22 - Equivalent Image Ellipse

$$a = \left(\frac{\mu_{2,0} + \mu_{0,2} + [(\mu_{2,0} - \mu_{0,2})^2 + 4\mu_{1,1}^2]^{1/2}}{\mu_{0,0}/2}\right)^{1/2} \quad (79)$$

$$b = \left(\frac{\mu_{2,0} + \mu_{0,2} - [(\mu_{2,0} - \mu_{0,2})^2 + 4\mu_{1,1}^2]^{1/2}}{\mu_{0,0}/2}\right)^{1/2} \quad (80)$$

$$\phi = (1/2)\tan^{-1}\left(\frac{2\mu_{1,1}}{\mu_{2,0} - \mu_{0,2}}\right) \quad (81)$$

Hence, if only these moments are known then only very general information about the image shape is available. However, even in this case enough information may be available to achieve some feature extraction. For example, it should be possible to distinguish long, thin features such as roads, airport runways

and shorelines from compact features such as buildings and vehicles. Long, thin objects are characterized by a>>b or b>>a while compact features are characterized by $a \sim b$. Rotating the image results in change in the angle ϕ. However, image rotation will not change a or b. Hence, a and b are invariant to rotation. Because they are formed from central moments they are also invariant to translation. And, finally, because of the $\mu_{0,0}$ term in the denominator they are invariant to intensity changes.

Since a and b are invariant to translation and rotation so are

$$a^2 + b^2 = \frac{2(\mu_{2,0} + \mu_{0,2})}{\mu_{0,0}/2} \tag{82}$$

$$a^2 - b^2 = \frac{2[(\mu_{2,0} - \mu_{0,2})^2 + 4\mu_{1,1}^2]^{1/2}}{\mu_{0,0}/2} \tag{83}$$

These last two moment invariants were used by Hu to distinguish between letters of the alphabet[12].

Moments can be modified to be invariant to scale. Consider the scale change $x' = \alpha x$, $y' = \alpha y$. Then it is straightforward to show that the unprimed central moments $\mu_{p,q}$ will change to the primed central moments $\mu'_{p,q}$ where

$$\mu'_{p,q} = \alpha^{(p+q+2)} \mu_{p,q} \tag{84}$$

To achieve invariance to scale first make use of the relationship

$$\mu'_{0,0} = \alpha^2 \mu_{0,0} \tag{85}$$

so that

$$\alpha^{(p+q+2)} = \left(\frac{\mu'_{0,0}}{\mu_{0,0}} \right)^{1+(p+q)/2} \tag{86}$$

Substituting this expression into equation (84) and separating primed from unprimed terms yields

$$\frac{\mu'_{p,q}}{(\mu'_{0,0})^{1+(p+q)/2}} = \frac{\mu_{p,q}}{\mu_{0,0}^{1+(p+q)/2}} \tag{87}$$

Hence,

$$\tilde{\mu}_{p,q} = \mu_{p,q}/\mu_{0,0}^{1+(p+q)/2} \qquad (88)$$

is invariant to scale. The moments $\tilde{\mu}_{p,q}$ will be termed scale-normalized moments.

The direct substitution of $\tilde{\mu}$ moments for μ moments in the expressions for a,b, a^2+b^2 and a^2-b^2 yields invariants $\tilde{a},\tilde{b}$, $\tilde{a}^2+\tilde{b}^2$ and $\tilde{a}^2-\tilde{b}^2$ respectively. These invariants are translation, rotation and scale invariant. Note, although a,b and $a^2\pm b^2$ are invariant to intensity variation, $\tilde{a},\tilde{b}$ and $\tilde{a}^2\pm\tilde{b}^2$ are not. To see this, let I(x,y) change to kI(x,y) then a,b and $a^2\pm b^2$ do not change while $\tilde{a}$ and $\tilde{b}$ change to $k^{-1/2}\tilde{a}$ and $k^{-1/2}\tilde{b}$ and $\tilde{a}^2\pm\tilde{b}^2$ changes to $k^{-1}(\tilde{a}^2+\tilde{b}^2)$. This is not necessarily a problem, it should be kept in mind when applying these formulas, however. It should be noted that the normalization given in equation (88) is not the only one possible. Another scale-normalized set of moments is given by [9,13]

$$\hat{\mu}_{p,q} = \mu_{p,q}/(\mu_{2,0}+\mu_{0,2})^{(2+p+q)/4} \qquad (89)$$

Up to this point moments and central moments have been defined. A few scalers have been given which are invariant to translation, rotation and scale. These results can be generalized. In references 12 and 13 it is shown that by including higher order moments, additional invariant scalers can be derived which are translation, rotation and scale invariant. Higher order moments contain more information about the image. In his paper[12] Hu was able to differentiate long, slender letters such as I or L from compact letters such as N or M using only $a^2\pm b^2$. However, he was not able to differenciate between letters which have about the same shape such as the pairs (W,M), (E,F) or (B,R). In a more recent paper[13] Teague has shown that moments up to at least the 11th or 12th order are needed to distinguish an E from an F.

Assume now that a sufficiently high number of moments are available so that m, invariant scaler functions of these moments could be computed. Then the m-vector of these scalers could serve as an invariant feature signature. This m-vector would then be the output of the prefilter. This approach has been used to identify aircraft, for example.[9]

Consider now the computation of image moments using A-O devices. Two approaches to this problem will be considered. They are the following:

1. Computing image moments from the two-dimensional Fourier transform. The Fourier transform is computed by the A-O device.

2. Computing image moments directly by modifying the A-O device input signals g(x) and h(x) in equation 1.

First, consider the approach requiring the Fourier transform. Some of the results in this development will be applicable to the second approach. Given that the two-dimensional Fourier transform is available, it is possible to compute image moments from samples of the Fourier transform. Two methods of accomplishing this will be considered.

The first method can be developed as follows. Start with the Fourier transform of the image.

$$F(\omega_x, \omega_y) = \iint I(x,y)e^{-j(\omega_x x+\omega_y y)}\, dxdy \tag{90}$$

Now expand the exponential in a power series and integrate term by term. The result is

$$F(\omega_x, \omega_y) = \sum_{p=0}^{\infty} \sum_{q=0}^{\infty} \frac{(-j)^{p+q}}{p!q!} m_{p,q}\omega_x^{\ p}\omega_y^{\ q} \tag{91}$$

so that

$$m_{p,q} = (-j)^{-(p+q)}\left[\left(\frac{\partial}{\partial \omega_x}\right)^p \left(\frac{\partial}{\partial \omega_y}\right)^q F(\omega_x, \omega_y)\right]_{\omega_x=\omega_y=0} \tag{92}$$

Hence, $m_{p,q}$ can be derived from the partial derivatives of F evaluated at the origin. This approach has been considered by Teague [29]. Since $F(\omega_x, \omega_y)$ and not its partial derivatives

are available, the partial derivates must be approximated by numerical differentiation. Partial derivatives are replaced by finite differences

$$\left(\frac{\partial}{\partial\omega_x}\right)^p \left(\frac{\partial}{\partial\omega_y}\right)^q F(\omega_x,\omega_y) \approx (\Delta_{\omega_x})^p (\Delta_{\omega_y})^q F(\omega_x,\omega_y) \tag{93}$$

where

$$\Delta\omega_x \; F(\omega_x,\omega_y) = \frac{F(\omega_x+h,\omega_y) - F(\omega_x-h,\omega_y)}{2h} \tag{94}$$

and

$$\Delta\omega_y \; F(\omega_x,\omega_y) = \frac{F(\omega_x,\omega_y+h) - F(\omega_x,\omega_y-h)}{2h} \tag{95}$$

It can be shown[29] that in order to measure all moments of order n or less (i.e., $p+q \leq n$) requires $2n^2 + 2n+1$ distinct samples of the transform. The location of these samples is shown in Figure 23. In order to compute moments of order n or less, the required samples (which are the dots) are all located on or within the square labeled n.

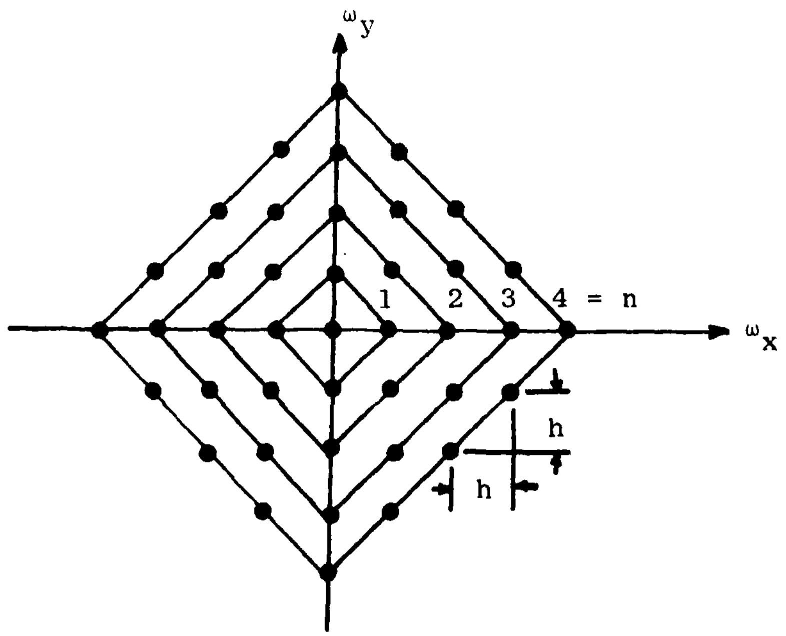


Figure 23 - Transform Sampling: Finite Differences

The only control on accuracy is the step size h. For high accuracy, h may be so small that all the samples are restricted to the area of the transform occupied by the d.c. sinc function. (The Fourier transform of an image will always contain a term centered at $\omega_x = \omega_y = 0$ due to the average light intensity of the image. This term is a square, two-dimensional sinc function because the A-O sensor is assumed to have a square aperture.) If this is the case then higher frequency components of the transform will not be utilized in the computation of the moments. While it is true that if the Fourier transform were known with infinite precision then decreasing h will always result in greater accuracy. The exact Fourier transform is an analytic function and so the entire transform can be reconstructed from the partial derivatives evaluated at $\omega_x = \omega_y = 0$. However, in practical A-O sensors, greater measurement accuracy can be achieved by utilizing Fourier transform data over the entire region of Fourier space which is available. The small effect of a high spatial frequency component on the partial derivatives at the origin will be error-prone whereas an accurate measurement at the high spatial frequency is possible. Hence, this method of computing moments does not appear well-suited for implementation using A-O devices since there is no practical means available for controlling accuracy of computation.

Because of this problem an alternative method was developed under this contract. This method also uses samples of the Fourier transform. However, the sample spacing is fixed and the accuracy of the method is adjusted by adjusting the number of samples. In this method, higher accuracy requires higher spatial frequency components. The method seems well-suited for A-O device implementation and it will now be developed.

The starting point in this development is the defining

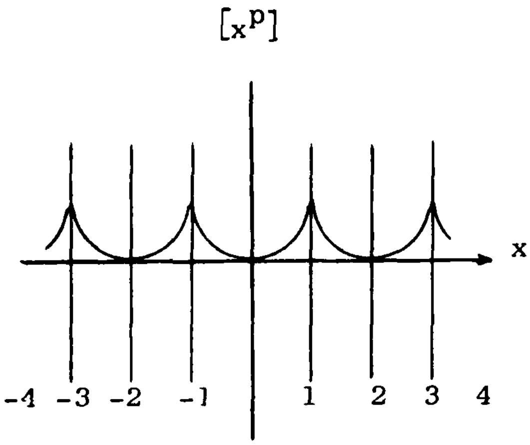
equation for two-dimensional moments, equation (67). Now the A-O device has a finite aperture. This aperture will be considered square. Because of this, the weight functions x^p, y^q are only required to have this form within the device aperture. Let these functions be periodically extended beyond the device aperture. This is shown for the one-dimensional case in Figure 24. The extension to two dimensions is straightforward. The device aperture is considered to be centered at x = 0 and to extend from x = -1 to x = +1. Four cases are shown in the figure. These are : a) p even and period = 2; b) p even and period = 4; c) p odd and period = 2; d) p odd and period = 8. The periodic extension of x^p is indicated with the notation $[x^p]$. The periodic extension of y^q is $[y^q]$. Over a single period, centered around x = 0, the functional definition of $[x^p]$ for the four cases shown in the figure are as follows:

p even and period = 2:
$$[x^p] = x^p;\ -1 \le x \le 1 \tag{96}$$

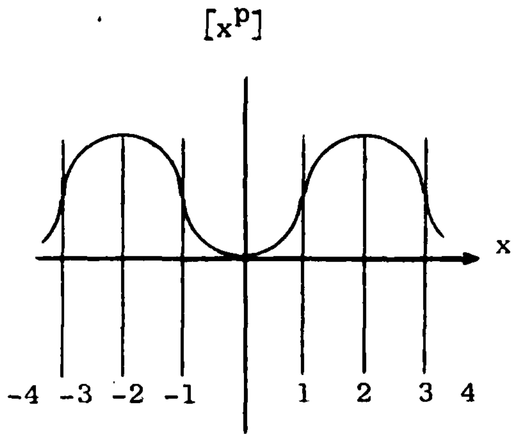
p even and period = 4:
$$[x^p] = \begin{cases} 2-(x+2)^p; & -2 \le x \le -1 \\ x^p; & -1 \le x \le +1 \\ 2-(x-2)^p; & +1 \le x \le +2 \end{cases} \tag{97}$$

p odd and period = 2:
$$[x^p] = x^p;\ -1 \le x \le 1 \tag{98}$$

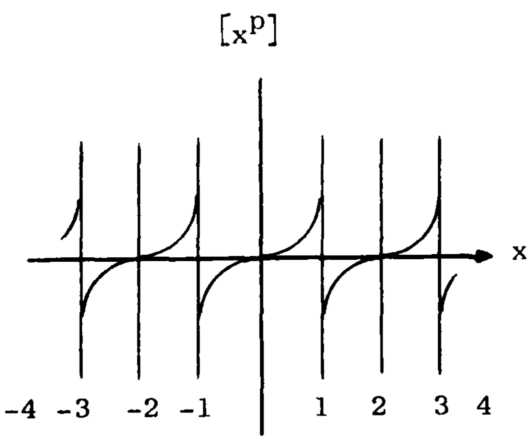
p odd and period = 8:
$$[x^p] = \begin{cases} -(x+4)^p & ;\ -4 \le x \le -3 \\ -2-(x+2)^p & ;\ -3 \le x \le -2 \\ -2+(x+2)^p & ;\ -2 \le x \le -1 \\ x^p & ;\ -1 \le x \le +1 \\ 2+(x-2)^p & ;\ +1 \le x \le +2 \\ 2-(x-2)^p & ;\ +2 \le x \le +3 \\ -(x-4)^p & ;\ +3 \le x \le +4 \end{cases} \tag{99}$$



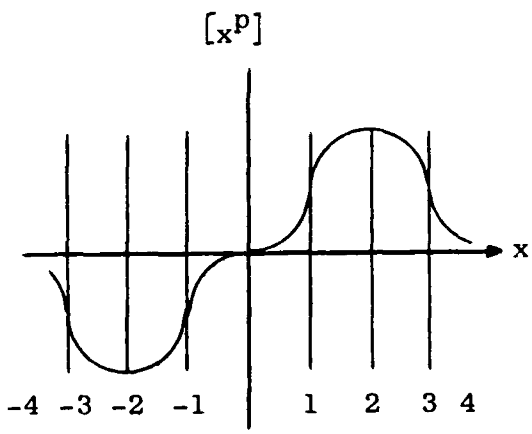
a. p even and period = 2



b. p even and period = 4



c. p odd and period = 2



d. p odd and period = 8

Figure 24 - Periodic Extensions of x^p

Since $[x^p]$ is periodic, it can be approximated by a finite sum of complex exponentials. There are a number of ways of achieving such an approximation. For example, the finite sum will be "best" in a mean squares sense if a truncated Fourier series is used.

An alternative method, the method of trigonometric interpolation, will be developed here[30]. This method has the advantage that simple formulas are available for determining the coefficients of the expansion. The method can therefore be easily modified to produce exponential sum approximations to any periodically extended function for which a closed form formula exists without the need for an integration. (An integration is usually required to determine Fourier series coefficients.) The method is, therefore, well-suited for computer generation of the required coefficients. Although the method is very general, the following development will be restricted to the special case of interest, namely the expansion of $[x^p]$.

The method has the property that the approximation to $[x^p]$ will exactly equal $[x^p]$ on a set of grid points

$$x_i = iL/n;\; i = 0, \pm 1, \pm 2, \ldots, \pm n \tag{100}$$

where L is one half the period of $[x^p]$. The number of points, n, is a parameter which can be set to achieve the required accuracy.

The approximations to $[x^p]$ and $[y^q]$ will be denoted by $g_p(x)$ and $h_q(y)$ respectively where

$$g_p(x) = \sum_{k=-n}^{n}{}' \; c_k(p) e^{j(k\pi x/L)} \tag{101}$$

$$h_q(y) = \sum_{k=-n}^{n}{}' \; c_k(q) e^{j(k\pi y/L)} \tag{102}$$

The notation $c_k(p)$, $c_k(q)$ for the coefficients is used to indicate that these are functions of the powers p and q.

The notation Σ' means

$$\sum_{k=-n}^{n}{}' a_k = \sum_{k=-n}^{n} a_k - 1/2(a_n+a_{-n}) \tag{103}$$

Now refer back to Figure 24. The function $[x^p]$ has different properties depending on whether p is even or odd and the period 2, 4 or 8. If p is even and period = 2 then $[x^p]$ is continuous but the first derivative is discontinuous. If p is even and period = 4 then both $[x^p]$ and its first derivative are continuous. If p is odd and period = 2 then $[x^p]$ is discontinuous. If p is odd and period = 8 then both $[x^p]$ and its derivative are continuous. If either $[x^p]$ or its derivative is discontinuous at $x = \pm 1$ the error in the approximation near these points will be rather large unless n is large. Therefore, it appears better to use the periodic extensions shown in Figure 24 b. and c. These extensions will be exclusively used throughout the remainder of this development. Notice that if p is odd the period must be twice as long as when p is even. To account for this, L in equations (101) and (102) will henceforth be denoted, when necessary, by L_e or L_o for the even or odd case respectively. If the normalization of Figure 24 is used then $L_e = 2$ and $L_o = 4$.

The formula for finding $c_k(p)$ given $[x^p]$ is given in reference 30. First, express the complex coefficient $c_k(p)$ in terms of its real and imaginary parts

$$c_k(p) = \frac{\alpha_k(p)-j\beta_k(p)}{2} \tag{104}$$

where

$$\alpha_k(p) = \frac{1}{n} \sum_{\ell=-n}^{n}{}' g_p(x_\ell)\cos(k\pi x_\ell/L) \tag{105}$$

$$\beta_k(p) = \frac{1}{n} \sum_{\ell=-n}^{n}{}' g_p(x_\ell)\sin(k\pi x_\ell/L) \tag{106}$$

and $x_\ell = \ell L/n$. The formulas for $c_k(q)$, $\alpha_k(q)$ and $\beta_k(q)$ are the same except that p is replaced by q, x is replaced by y

and g_p is replaced by h_q everywhere they occur. Since the following development is identical for both g_p and h_q, only the case for g_p will be shown.

The function $g_p(x)$ equals $[x^p]$ at all of the 2n+1 points x_ℓ (and also at all periodic extensions of these points). There will generally be some error between the mesh points. This will be investigated later.

Now g_p is either an even or an odd function depending on whether p is even or odd. For p even

$$c_k(p) = 1/2\, \alpha_k(p) \tag{107}$$

for p odd

$$c_k(p) = -1/2j\, \beta_k(p) \tag{108}$$

For p even or odd L_e or L_o is used. To obtain equivalent accuracy, the mesh space should be the same in both cases. Since $L_o = 2L_e$ the number, n, of mesh points will be different in each case. Denote them as n_e and n_o for even and odd cases respectively. The formula for $c_k(p)$ and p even is then given by

$$\text{p even: } c_k(p) = \frac{1}{2n} \sum_{\ell=-n_e}^{n_e}{}' g_p(x_\ell)\cos(k\pi x_\ell/L_e) \tag{109}$$

$$= \begin{cases} 1; & p = 0 \text{ and } k = 0 \\ 0; & p = 0 \text{ and } k \neq 0 \\ \frac{1}{n} \sum_{\ell=1}^{n_e}{}' g_p(x_\ell)\cos(k\pi x_\ell/L_e); & p>0 \end{cases} \tag{110}$$

and for p odd,

$$\text{p odd: } c_k(p) = -\frac{j}{2n} \sum_{\ell=-n_o}^{n_o}{}' g_p(x_\ell)\sin(k\pi x_\ell/L_o) \tag{111}$$

$$= -\frac{j}{n} \sum_{\ell=1}^{n_o}{}' g_p(x_\ell)\sin(k\pi x_\ell/L_o) \tag{112}$$

where the notation Σ' here means

$$\sum_{\ell=1}^{n}{}' a_\ell = \sum_{\ell=1}^{n} a_\ell - 1/2\, a_n \tag{113}$$

If p is even then $g_p(x_\ell)\cos(k\pi x_\ell/L_e)$ is an even function. Likewise, if p is odd then $g_p(x_\ell)\sin(k\pi x_\ell/L_o)$ is an odd function. Because of this

$$c_k(p) = c_{-k}(p);\ p \text{ even} \tag{114}$$

$$c_k(p) = -c_{-k}(p);\ p \text{ odd} \tag{115}$$

These expressions reduce the number of multiplies required to compute $g_p(x)$ from equation (101) by a factor of two.

There is an additional symmetry present in $[x^p]$ which reduces the number of multiplies still further. The cases of p even and odd must be considered separately.

First, consider the case where p is even and p>0. Consider the function

$$\hat{g}_p(x) = g_p(x-1) - 1;\ p>0 \text{ and even} \tag{116}$$

in the interval $-1 \leq x \leq +1$ the function $\hat{g}_p(x)$ is an odd function while the function $\cos(k\pi(x-1)/L_e)$ is even in this same interval for k even and k>0. Hence, there will be no contribution to $c_k(p)$ from the points in this interval when k>0 and even. An examination of the function

$$\tilde{g}_p(x) = g_p(x+1) - 1;\ p>0 \text{ and even} \tag{117}$$

in the interval $-1 \leq x \leq +1$ leads to the same result. Hence, for p>0 and even and k>0 and even $c_k(p) = 0$.

Now consider the case the p is odd. Consider the function

$$\bar{g}_p(x) = g_p(x-2);\ p \text{ odd} \tag{118}$$

in the interval $-2 \leq x \leq +2$. The function $\bar{g}_p(x)$ is even in this interval while the function $\sin(k\pi(x-2)/L_o)$ is odd in this same interval for k even and k>0. Hence, there will be no contribution to $c_k(p)$ from points in this interval when k>0

and even. An examination of the function

$$\bar{\bar{g}}_p(x) = g_p(x+2); \text{ p odd} \tag{119}$$

in the interval $-2 \leq x \leq +2$ leads to the same result. Hence, for p odd and k>0 and even, $c_k(p) = 0$. In summary:

$$c_k(p) = c_{-k}(p); \text{ p even} \tag{120}$$

$$c_k(p) = c_{-k}(p); \text{ p odd} \tag{121}$$

$$c_k(p) = 0; \text{ p even and } k>0 \text{ and even} \tag{122}$$

$$c_k(p) = 0; \text{ p odd and } k>0 \text{ and even} \tag{123}$$

Now consider the generalization to two-dimensions of the results to this point. Let $[x^p][y^q]$ be the periodic extension of $x^p y^q$ outside the square with corners (-1,-1), (-1,1), (1,-1), (1,1). Then $[x^p][y^q]$ can be approximated by $g_p(x)h_q(y)$ where

$$g_p(x)h_q(y) = \sum_{i=-n}^{n}{}' \sum_{k=-n}^{n}{}' c_i(p)c_k(q)e^{j\pi(ix/L+ky/L)} \tag{124}$$

where the various L's and n's are to be replaced by L_e, L_o, n_e, n_o - whichever is appropriate. Now consider the defining equation for $m_{p,q}$ equation (67). If the limits of integration are restricted to $-1 \leq x,y \leq +1$ then the kernal $x^p y^q$ can be replaced by $[x^p][y^q]$ without effecting the result. That is,

$$m_{p,q} = \int_{-1}^{1} \int_{-1}^{1} [x^p][y^q] I(x,y)dxdy \tag{125}$$

Now approximate $[x^p][y^q]$ by $g_p(x)h_q(y)$

$$m_{p,q} \approx \int_{-1}^{1} \int_{-1}^{1} g_p(x)h_q(y)I(x,y)dxdy \tag{126}$$

Now substitute the defining trigonmetric formulas for g_p and h_q and interchange integration and summation. The result is

$$m_{p,q} \approx \sum_{i=-n}^{n}{}' \sum_{k=-n}^{n}{}' c_i(p)c_k(q) \int_{-1}^{1} \int_{-1}^{1} I(x,y)e^{j\pi(ix/L+ky/L)}\, dxdy \tag{127}$$

$$= \sum_{i=-n}^{n}{}' \sum_{k=-n}^{n}{}' c_i(p)c_k(q)F(-\frac{i\pi}{L},-\frac{k\pi}{L}) \qquad (128)$$

Since I(x,y) is real $F(\omega_x,\omega_y) = F^*(-\omega_x,-\omega_y)$

so that

$$m_{p,q} \approx \sum_{i=-n}^{n}{}' \sum_{i=-n}^{n}{}' c_i(p)c_k(q)F^*(\frac{i\pi}{L},\frac{k\pi}{L}) \qquad (129)$$

this is the desired result. It states that if $x^p y^q$ are approximated over the aperture $-1 \le x,y \le +1$ by the trigonometric series $g_p(x)h_q(y)$ with coefficients $c_i(p)$ and $c_k(q)$ then $m_{p,q}$ is approximated by the weighted double sum of two-dimensional Fourier transform samples with the same coefficients $c_i(p)$ and $c_k(q)$. Notice that when a larger value of n is used in order to increase accuracy then the Fourier transform must be sampled over a larger region of Fourier space.

Now, taking into account the symmetries present in the coefficients $c_i(p)$ and $c_k(q)$ and in the two-dimensional Fourier transform equation (129) can be simplified. First of all, define the notation

$$\sum_{i=0}^{n}{}' a_i = \sum_{i=0}^{n} a_i - 1/2(a_o+a_n) \qquad (130)$$

then the summation approximating $m_{p,q}$ can be broken into four parts.

$$m_{p,q} \approx \sum_{i=0}^{n}{}' \sum_{k=0}^{n}{}' c_i(p)c_k(q)F^*(\frac{i\pi}{L},\frac{k\pi}{L})$$

$$+ \sum_{i=0}^{n}{}' \sum_{k=0}^{-n}{}' (\text{same}) + \sum_{i=0}^{-n}{}' \sum_{k=0}^{n}{}' (\text{same}) + \sum_{i=0}^{-n}{}' \sum_{k=0}^{-n}{}' (\text{same}) \qquad (131)$$

Since I(x,y) is real $F(\omega_x,\omega_y) = F^*(-\omega_x,-\omega_y)$ so that

$$m_{p,q} \approx \sum_{i=0}^{n}{}' \sum_{k=0}^{n}{}' [c_i(p)c_k(q)F^*(\frac{i\pi}{L},\frac{k\pi}{L})+c_{-i}(p)c_{-k}(q)F(\frac{i\pi}{L},\frac{k\pi}{L})]$$

$$+ \sum_{i=0}^{n}{}' \sum_{k=0}^{n}{}' [c_i(p)c_{-k}(q)F^*(\frac{i\pi}{L},\frac{-k\pi}{L})+c_{-i}(p)c_k(q)F(\frac{i\pi}{L},\frac{-k\pi}{L})] \qquad (132)$$

Now $F+F^* = 2$ real (F), $F-F^* = 2j$ imag(F) and taking advantage of the symmetries equations (120), (121), (122) and (123) the formula for $m_{p,q}$ can be broken into four pieces for the four cases

1. p and q even
2. p even and q odd
3. p odd and q even
4. p and q odd

These cases are the following:

<u>case 1: p and q even</u>

$$m_{p,q} = 2 \sum_{i=0}^{n_e}{}' \sum_{k=0}^{n_e}{}' c_i(p)c_k(q)\left[\text{real } F(i\pi/L_e, k\pi/L_e) + \text{real } F(i\pi/L_e, -k\pi/L_e)\right] \tag{133}$$

<u>case 2: p even and q odd</u>

$$m_{p,q} = 2j \sum_{i=0}^{n_e}{}' \sum_{k=0}^{n_o}{}' c_i(p)c_k(q)\left[\text{imag } F(i\pi/L_e, -k\pi/L_o) - \text{imag } F(i\pi/L_e, k\pi/L_o)\right] \tag{134}$$

<u>case 3: p odd and q even</u>

$$m_{p,q} = -2j \sum_{i=0}^{n_o}{}' \sum_{k=0}^{n_e}{}' c_i(p)c_k(q)\left[\text{imag } F(i\pi/L_o, k\pi/L_e) + \text{imag } F(i\pi/L_o, -k\pi/L_e)\right] \tag{135}$$

<u>case 4: p and q odd</u>

$$m_{p,q} = 2 \sum_{i=1}^{n_o}{}' \sum_{i=1}^{n_o}{}' c_i(p)c_k(q)\left[\text{real } F(i\pi/L_o, k\pi/L_o) - \text{real } F(i\pi/L_o, -k\pi/L_o)\right] \tag{136}$$

In all four cases, approximately three quarters of the combined coefficients $c_i(p)c_k(q)$ are zero since $c_i(p) = 0$ when $i>0$ and even and $c_k(q) = 0$ when $k>0$ and even. In all cases, $m_{p,q}$ is a real number. If p is even and q is odd or

p is odd and q is even then $c_i(p)c_k(q)$ is imaginary and $jc_i(p)c_k(q)$ is real. Sample spacing in the (ω_x, ω_y) - plane for the four cases above is shown in Figure 25. The dashed half square is the first zero of the two-dimensional sinc function which results from the average light intensity of the image. The transform is to be sampled at the intersection of all solid lines including the ω_x and ω_y-axis. The total number of samples depends on n_e and n_o. Notice that real or imaginary parts of the transform for samples which have the same ω_x- coordinate but opposite ω_y- coordinates are added or subtracted before being multiplied by $c_i(p)c_k(q)$.

Two Fortran computer programs have been written to compute the $c_i(p)$, $c_k(q)$ coefficients as defined by equations (105), (106), 107) and (108). These programs are listed in Appendix C.

The final topic which must be considered is how n_e and n_o are chosen to achieve a required accuracy in the approximations for $[x^p]$, $[y^q]$. To this end, two additional Fortran computer programs were written to analyze the error function

$$e_{p,n}(x) = [x^p] - g_p(x) \tag{137}$$

This is a function of x, the power p and the number of terms, 2n+1, in the summation defining $g_p(x)$, equation (101). Now because of the form of $g_p(x)$

$$e_{p,n}(x_i) = 0 \quad i = 0, \pm 1, \pm 2, \ldots \pm n; \; x_i = iL/n \tag{138}$$

Because of the symmetries present it is easy to see that

$$|e_{p,n}(x)| = |e_{p,n}(-x)| \tag{139}$$

Hence, it is only necessary to examine $e_{p,n}$ for positive values of x. If p is even then we are only interested in $e_{p,n}$ over the interval $0 \leq x \leq x_{n/2}$ which is right-half of the sensor aperture (normalized). If p is odd then we are, instead, interested in $e_{p,n}$ over the interval $0 \leq x \leq x_{n/4}$. The Fortran

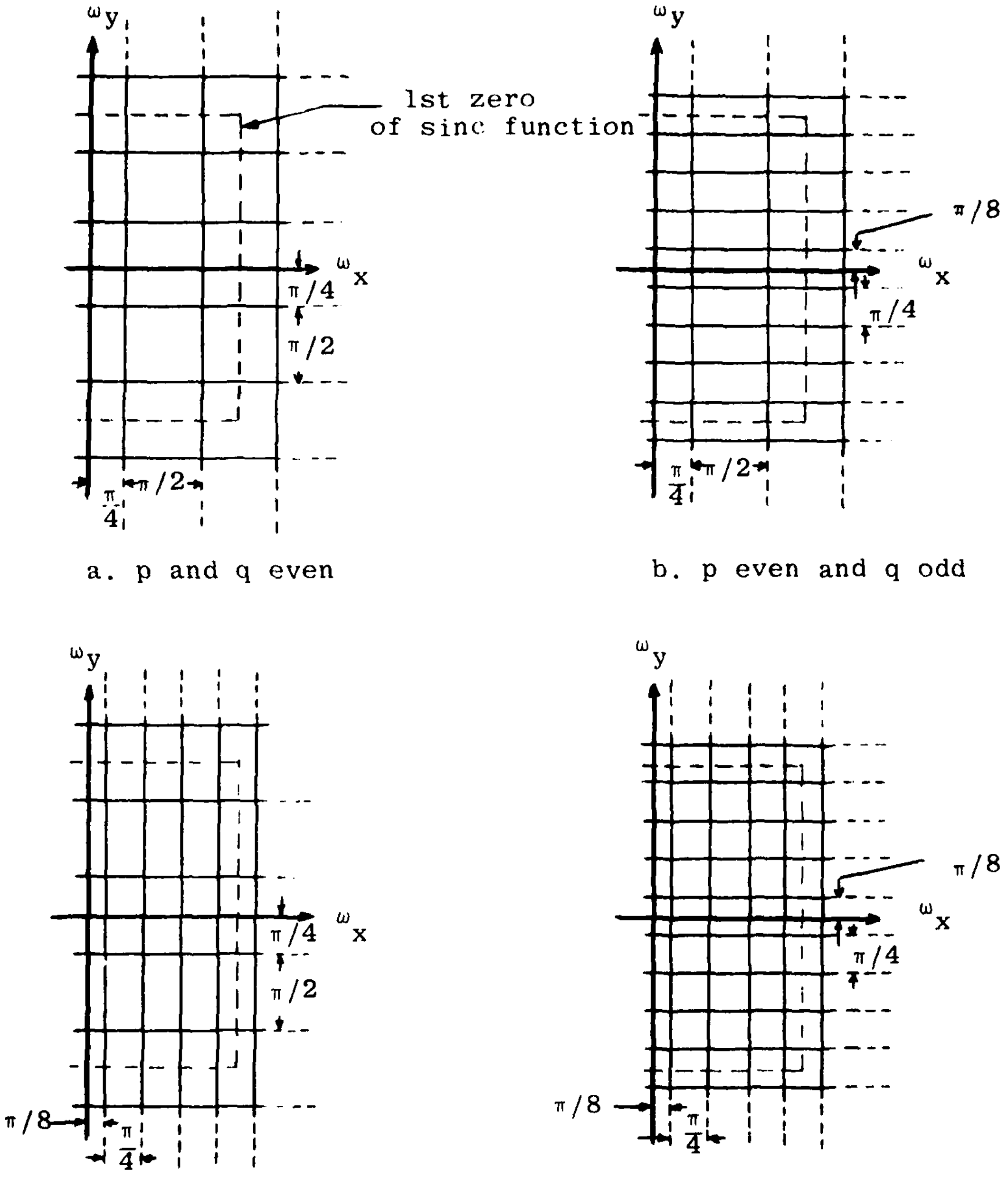


Figure 25 - Fourier Transform Sample Spacing for Moment Computation

programs allow the user to enter p, n/2 or n/4 and a mesh divisor m. The program then computes all the $c_i(p)$ coefficients and evaluates $e_{p,n}(x)$ at the points

$$x_i = iL/nm;\ i = 0,1,\ldots,\ nm/n_p \tag{140}$$

where $n_p = 2$ if p is even and $n_p = 4$ if p is odd. That is, the interval $x_i \leq x \leq x_{i+1}$ is divided into m parts.

Using these Fortran programs a study was conducted to determine the effect of n on the maximum error observed at the $1+nm/n_p$ grid points. The mesh divisor was fixed at m = 10 for all cases. Then p and n were varied and the maximum error overall the grid points observed and tabulated. For the cases p = 1 and n/4 = 1, 2 and 4 the error function is shown in Figure 26. Notice that the maximum error decreases by at least a factor of two for every doubling of n. The tabulation of maximum error for the cases of $1 \leq p \leq 8$ and n/n_p = 1,2,4 and 8 is given in Table 3. From this table it is seen that the maximum error increases for n/n_p fixed and increasing p. Likewise, if p is held constant then the maximum error decreases with increasing n/n_p. Now n/n_p is the number of points in the interval $0 < x \leq 1$ for which $e_{p,n}(x) = 0$. Notice that (approximately) the maximum error decreases by a factor of two when n/n_p is doubled and p held fixed. To investigate further, addition data was generated and tabulated in Table 4. From this table it is seen that the maximum error $|e_{p,n}|$ will remain approximately the same if when p is doubled n/n_p is also doubled. (More accurately, the maximum error is increasing slightly as p and n/n_p are progressively doubled.) Also note that for (approximately) the same error, n/n_p should be the same for p even and p odd. Since $n_p = 2$ for p even and $n_p = 4$ for p odd this implies that

$$n_o = 2n_e \tag{141}$$

for (approximately) the same error. A maximum $|e_{p,n}| = .03$ is an error of 3% of the full-scale value of $|x^p|$ in the interval $0 \leq x \leq 1$ which is 1.0.

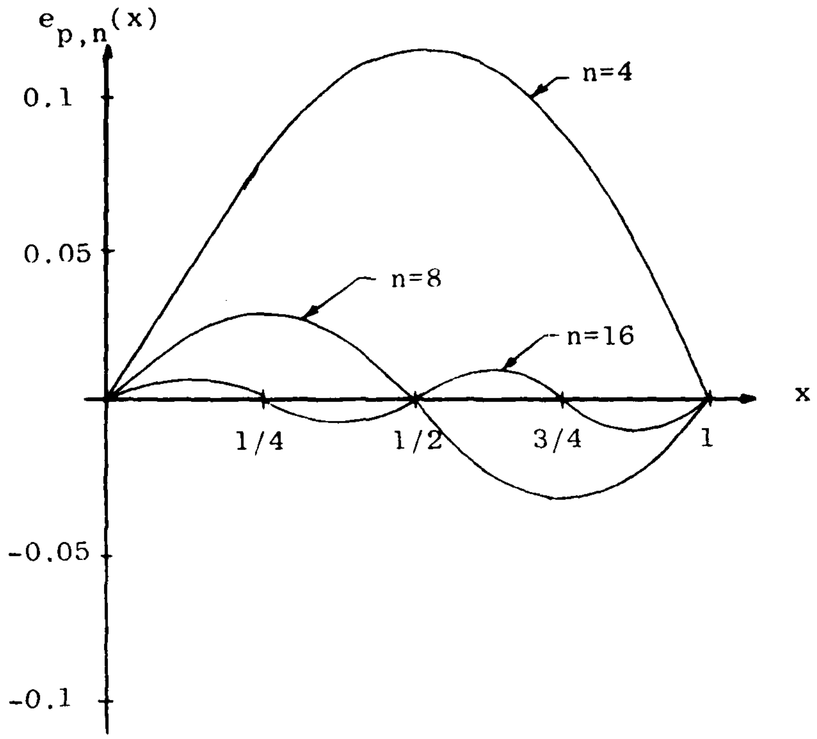


Figure 26 - $e_{p,n}(x)$ for $p = 1$ and $n/4 = 1,2$ and 4

p \ n/n_p	1	2	4	8
1	.1173166	.0353821	.0098962	.0026195
2	.0560095	.0172765	.0045921	.0011702
3	.2697230	.0526488	.0138003	.0035090
4	.3059095	.1007140	.0273291	.0070052
5	.4319005	.1697540	.0457383	.0116845
6	.4288390	.2163066	.0656872	.0173221
7	.5176162	.2795536	.0905453	.0241607
8	.5232109	.3136496	.1143713	.0317055

Table 3 Maximum $|e_{p,n}(x)|$ vs. p and n/n_p

Now refer to Figure 25 in order to relate n/n_p, and n to the spatial frequency bandwidth of the acousto-optic sensor. The first zero of the average intensity sinc function is the location of spatial frequencies of one line pair per aperture. The intersection of this zero with the ω_x and ω_y- axis is

$$\omega_x = \omega_y = \pi \tag{142}$$

A sensor with bandwidth of n_b line pairs per aperture has usable bandwidth

$$0 \leq \omega_x \leq n_b \pi$$

$$-n_b \pi \leq \omega_y \leq n_b \pi$$

Now $n_e = 2n/n_p$ and $n_o = 4n/n_p$. The bandwidth required for $g_p(x)$ is

$$\text{p odd:} \quad n_o \pi/8 = (4n/n_p)\pi/8 = (n/n_p)\pi/2 \tag{143}$$

$$\text{p even:} \quad n_e \pi/4 = (2n/n_p)\pi/4 = (n/n_p)\pi/2 \tag{144}$$

Equating with $n_b \pi$ yields

$$n/n_p = 2n_b \tag{145}$$

Referring to Table 4, in order to compute up to 16th order moments with maximum $|e_{p,n}| \leq .0336$ requires $n/n_p = 16$. Equation (145) then implies that the sensor bandwidth must be at least 32 line pairs per aperture along both the ω_x and ω_y -axis. Continuing with this example, if the sensor bandwidth was 16 line pairs per aperture then only 8th order moments could be computed with the same maximum $|e_{p,n}|$. However, if the error criteria was relaxed to maximum $|e_{p,n}| \leq .0672$ then 16th order moments could be computed with a 16 line pair per aperture device. Table 3 and 4 or the computer programs of Appendix C can be used to determine the bandwidth required

p	n/n_p	$\max\|e_{p,n}\|$
2	2	.0172765
4	4	.0273291
8	8	.0317055
16	16	.0335946

a. p even

p	n/n_p	$\max\|e_{p,n}\|$
1	2	.0353821
3	4	.0138003
7	8	.0241607
15	16	.0296186

b. p odd

Table 4 - Maximum $|e_{p,n}(x)|$ vs. $p = n/n_p$ or $p = n/n_p - 1$

given the highest order moments to be computed along with the largest allowable error. This analysis has assumed that the Fourier transform can be computed with perfect accuracy. Errors in the transform will lead to additional errors in the computed moments.

Since the two-dimensional moments are computed as weighted sums of two-dimensional Fourier transform components, the implementation of a preprocessor using an A-O device can take the form of Figure 21. The A-O device is used to compute Fourier components and the weighting and summation is carried out in the digital processor. The digital processor is also used to compute the moment invariants. An advantage of this method of computing moment invariants is that the same hardware configuration can be used to compute moment invariants as well as to compute the invariant signatures developed in Section III. 3.C. This would be desirable in applications where a decision is to be made based on prefilter outputs derived from more than one algorithm.

The algorithm for computing moments from the Fourier transform can be modified to provide some noise filtering. In aerial images it is often the case that the features of a signal that permit discrimination generally have significant high spatial frequency content in some frequency band[17]. The noise contributes primarily to spatial frequencies outside this band. It may be desirable to weight the Fourier transform components to emphasize signal and de-emphasize noise. The algorithm permits a weight function of the form

$$W(\omega_x,\omega_y) = W_x(\omega_x)W_y(\omega_y) \tag{146}$$

to be applied to the Fourier transform components prior to summation. These weights should be combined with the coefficients $c_i(p)$, $c_k(q)$. The composite coefficients are

$$\tilde{c}_i(p) = W_x(i\pi/L)c_i(p) \tag{147}$$

$$\tilde{c}_k(q) = W_y(k\pi/L)c_k(q) \tag{148}$$

These multiplications can be precomputed and stored. Hence, noise reduction can be added to the algorithm without the need for any additional on-line computations. Examples of W_x functions are

$$W_x(\omega_x) = \begin{cases} 0; & \omega_x < \omega_c \\ 1; & \omega_x \geq \omega_c \end{cases} \tag{149}$$

which is a high-pass filter and

$$W_x(\omega_x) = \omega_x \tag{150}$$

which is a derivative filter. It should be noted however that two-dimensional filters such as

$$W_x W_y = \omega_x \omega_y \tag{151}$$

have a directional bias. This is not totally consistant with the desire to compute invariant signatures. This topic needs further investigation.

Next, an alternative approach to computing image moments will be considered. This approach utilizes the development based on the method of trigonometric interpolation which was derived above.

The A-O devices which have been considered in this report which compute the function given by equation 1 are linear in g and h so that

$$\iint \hat{I}(x,y) \sum_{i=0}^{n} \sum_{k=0}^{n} g_i(t-x/v_x) h_i(t-y/v_y)\, dxdy =$$

$$\sum_{i=0}^{n} \sum_{k=0}^{n} \iint \hat{I}(x,y) g_i(t-x/v_x) h_i(t-y/v_y)\, dxdy \tag{152}$$

This property allows an alternative means of computing moments. Rather than weighting and summing Fourier components in the frequency domain, the eigenfunctions can be weighted and summed in the time domain. The composite function is then applied to the A-O device. The A-O device output is then the

desired moment. The weighted eigenfunctions are given by equations (101) and (102). However, these are just the bandlimited approximations to $[x^p]$ and $[y^q]$ derived from the method of trigonometric interpolation. Hence, if the A-O device electrical inputs are the approximations to $[x^p]$ and $[y^q]$ then the device output will approximate $m_{p,q}$. This statement needs two qualifications. First, the device output will only equal $m_{p,q}$ at the time instant when the SAW or BAW are aligned as shown in Figure 24. That is, there is only one instant per period when the acoustic waves line up on the sensor to give the correct weighting function. The second qualification arises because of the image sampling caused by the metal grid pattern on the sensor. This was discussed in Section II. The origin of Fourier space is translated by $f_{o,x}$ and $f_{o,y}$. To likewise translate the spectra of $g_p(x)$ and $h_q(y)$ they must be multiplied by $e^{j2\pi f_{x,o}}$ and $e^{j2\pi f_{y,o}}$ respectively. In an actual implementation, the complex exponentials would be replaced with real sinusoids and a synchronous detector used to preserve the phase information. Such a circuit is described in reference 18. Hence, the sensor drive signals should be of the form

$$g(t) = g_p(t)\sin 2\pi f_{x,o}t \tag{153}$$

$$h(t) = h_q(t)\sin 2\pi f_{y,o}t \tag{154}$$

Now since g_p and h_q have been designed to be bandlimited, g and h will also be bandlimited and n can be chosen to achieve maximum accuracy given the A-O device bandwidth. Hence, all the previous analysis can be applied to this implementation also. The functions, equations (153) and (154) can be generated either by performing an analog multiplication of g_p or h_q with $\sin 2\pi ft$ or by precomputing time samplesof g(t) and h(t) and storing these samples in a fast digital memory. The memory would then be read, D/A converted and the analog samples smoothed and applied to the A-O sensor. The sensor

output should be sampled in synchronism with the input so that this output is sampled and held at the instant when the correct portion of $g_p(x)h_q(y)$ is on the active sensor area. The implementation of such a preprocessor is shown in Figure 27.

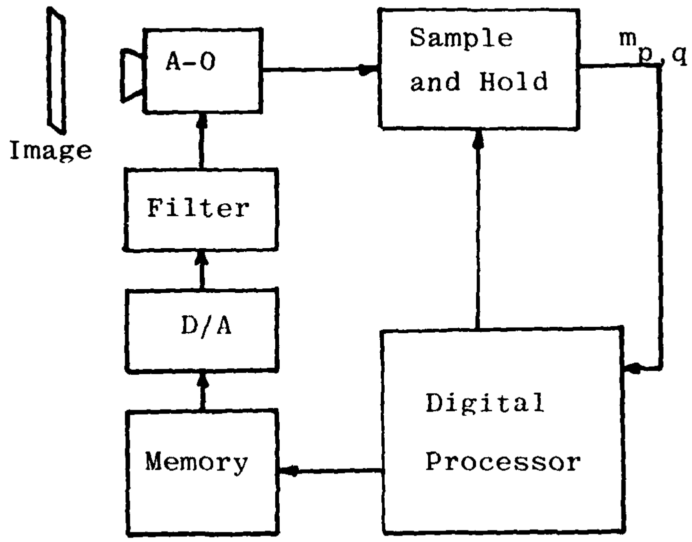


Figure 27 - Implementation: Method of Moments

D. Summary

This section began with a general discussion of prefilters. It was shown that the desirable property of feature isolation could be achieved by breaking the image into a number of smaller, overlapping views. By this means signal-to-noise ratio can be improved. Since A-O sensors cannot match the accuracy of a digital processor, feature isolation is useful since it relaxes the noise filtering requirements of the prefilter algorithm. However, the price paid is the need to process a number of views rather than a single aerial image.

The concept of feature invariance was introduced. A prefilter which is invariant to changes in the feature such

as translation or rotation or scale is very desirable. This is because the number of distinct signals at the prefilter output is greatly reduced which simplifies the decision processor. However, invariance usually leads to an undesirable reduction in separation between feature classes.

Because of the function performed by A-O sensors they are best suited to implement prefilters which utilize a separable transformation of the image of the form of equation (1). A number of such transformations were investigated.

The Hadamard transform is characterized as an orthonormal transform for which all the elements of the n x n transformation matrix H_n are either + or -1. Because of this property, it is well-suited for digital implementation since additions rather than multiplications are required. Since the A-O sensors performs general multiplications as easily as multiplications by ± 1, this special property of H_n is of no advantage in an A-O implementation. Since A-O devices are bandlimited whereas Walsh functions are not bandlimited, A-O devices do not appear to be the most optimum means for computing Hadamard transforms.

A classical means for feature extraction is the matched filter. However, the A-O implementation of the matched filter is not desirable since both forward and inverse Fourier transforms are required. The matched filter for the image autocorrelation function leads to a much simpler A-O implementation since only a forward transform is required. This version of the matched filter is invariant to feature translation but not rotation or scale.

More complex algorithms were developed which utilize Fourier transform samples to compute feature signatures which are invariant to translation, rotation and scale. Since the transform is sampled, the amount of information in the image is reduced to the point where a modest digital processor can compute the remaining steps in the algorithm. These algorithms are well suited for an A-O device implementation consisting of

an A-O sensor and a digital post processor.

Image two-dimensional moments were introduced. Certain combinations of these moments can be invariant to feature translation, rotation and scale change. Two methods of computing these moments from the Fourier transform were presented. The second method is more accurate for A-O implementation since it utilizes Fourier samples over the entire bandwidth of the sensor. An alternative method of computing moments by time-weighting the electrical inputs to the A-O sensor was developed. Using this method, moments can be computed potentially much faster than by the Fourier transform method. However, a more complex hardware configuration is required.

In summary, there appear to be a number of potentially useful feature extraction algorithms which can be effectively implemented using A-O sensors.

IV. FEATURE EXTRACTION EXPERIMENTS WITH DEFT SENSORS

A. Introduction

For the purpose of sensor and algorithm evaluation, Deft Laboratories Inc. has developed a microprocessor-based Deft sensor operating system. This system consists of a MC6800-based microprocessor system, two digitally controllable sine wave generators, a Deft sensor and an electronics module for signal filtering and amplification. The system can interface with a tape recorder, teletype, storage CRT and an X-Y plotter. Assembly language programs can be written to implement signal processing algorithms. In addition, some resident software is available to make measurements of the Deft sensor output and to make pseudo-three dimensional plots of the magnitude of the Fourier transform.

Since this facility is available, it was decided to program one or more of the prefilter algorithms under study in order to make a preliminary evalution of the concept of implementation developed in Section III. The present experimental set-up has two limitations. First, the system is limited to evaluating Deft sensors. Second, the sensor drive functions $g(t)$ and $h(t)$ are limited to sinusoids. Because of this second limitation experiments were restricted to those algorithms which characterize the image by its two-dimensional Fourier transform. Algorithms which were programmed are the method of invariant Fourier signatures and the method of invariant moment signatures. However, because of a shortcoming of the Deft sensors which were available during this study, it was not possible to compute image moments with any accuracy. This shortcoming is presently being eliminated in a new sensor design. More will be said about present shortcomings in the next section. The method of invariant Fourier signatures is less sensitive to this problem so that a series of experiments were conducted. The results of these experiments are detailed here.

First, however, the computer programs which were written will be described. This will be followed by a description of the experiments which were conducted and the results of these experiments. The section closes with some conclusions.

B. Feature Extraction Experimental Computer Programs

1. Method of Invariant Moment Signatures

The flow diagram of this program is rather straightforward and is shown in Figure 28. When the program is entered it requests that the sensor view a uniform image for calibration. When this image is in place the user types "C" and the program makes a series of measurements near the transform origin for the purpose of removing a linear phase term which is present in the Deft sensor output. This phase function is approximately of the form

$$\phi(\omega_x, \omega_y) = k_x\omega_x + k_y\omega_y + k_o \tag{155}$$

The computed Fourier transform is

$$\tilde{F}(\omega_x, \omega_y) = F(\omega_x, \omega_y)e^{j\phi(\omega_x, \omega_y)} \tag{156}$$

where F is the desired Fourier transform.

This phase function is undesirable. The constant k_o is measured at the Fourier origin and removed from all data points. The linear phase also must be measured and its effects removed from each data point. The method of trigonometric interpolation is implemented in this program. Refer to Figure 24. If linear phase is not removed then the effect is to shift the aperture of the sensor out of the normalized aperture centered at $x = y = 0$ and extending to $x = \pm 1, y = \pm 1$. The image will then, in effect, be multiplied by $[x^p]$, $[y^q]$ outside the interval where these functions equal x^p, y^q. The result will no longer be the image moment. To remove the linear phase, k_x and k_y are measured by sampling the transform at two points on the ω_x-axis and two points on the ω_y-axis near the Fourier origin where the transform has a large magnitude. Since a

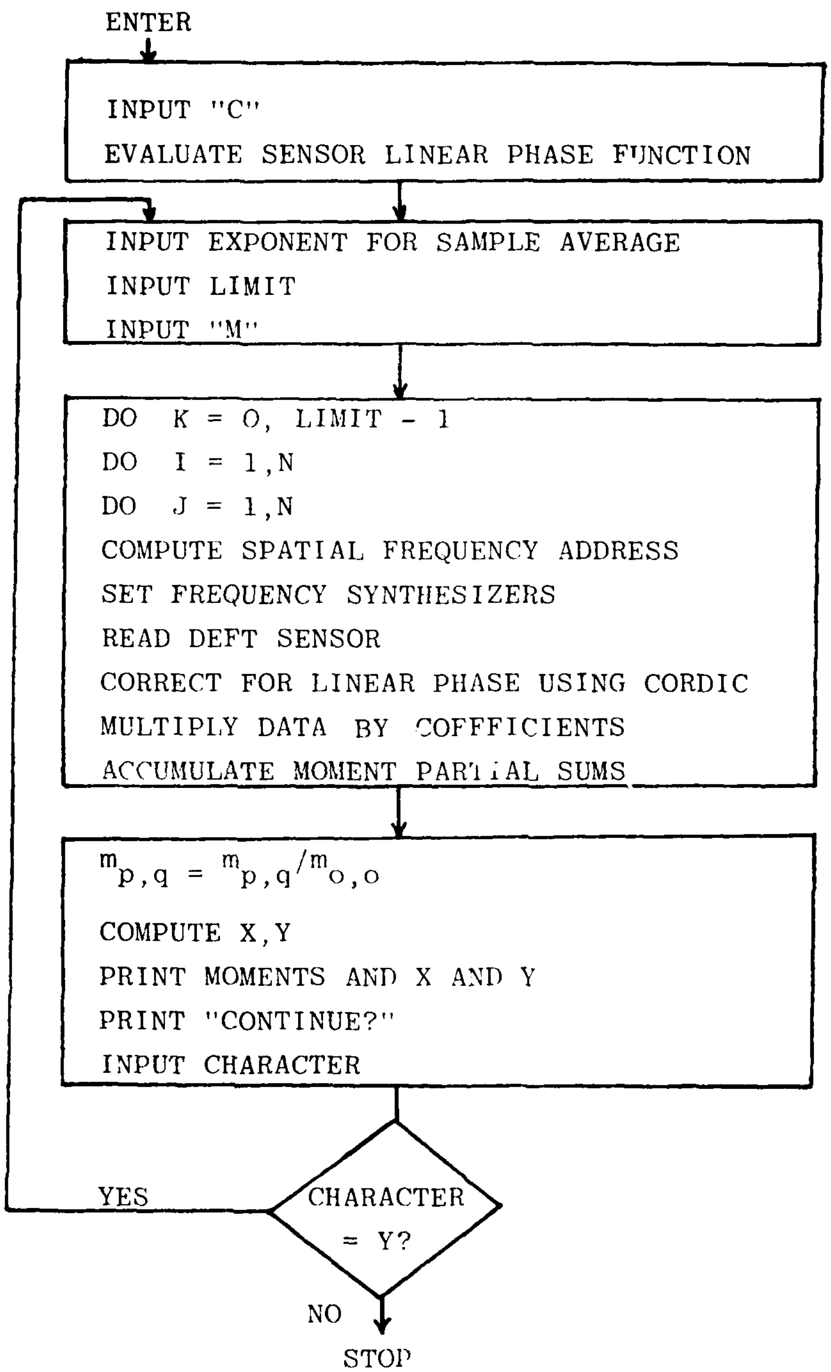


Figure 28 - Flow Diagram: Invariant Moment Signatures

uniform image has a transform which is a real function, any linear phase measured is due to the sensor and its associated electronics and should be removed. For example, if the sensor is measured at (o,o), $(\tilde{\omega}_x, 0)$ and $(o, \tilde{\omega}_y)$ with result

$$\tilde{F}(0,0) = A(0,0)e^{\,j\phi(0,0)} \tag{157}$$

$$\tilde{F}(\tilde{\omega}_x,0) = A(\tilde{\omega}_x,0)e^{\,j\phi(\tilde{\omega}_x,0)} \tag{158}$$

$$\tilde{F}(0,\tilde{\omega}_y) = A(0,\tilde{\omega}_y)e^{\,j\phi(0,\tilde{\omega}_y)} \tag{159}$$

then

$$k_o = \phi(0,0) \tag{160}$$

$$k_x = \frac{\phi(\tilde{\omega}_x,0)-\phi(0,0)}{\tilde{\omega}_x} \tag{161}$$

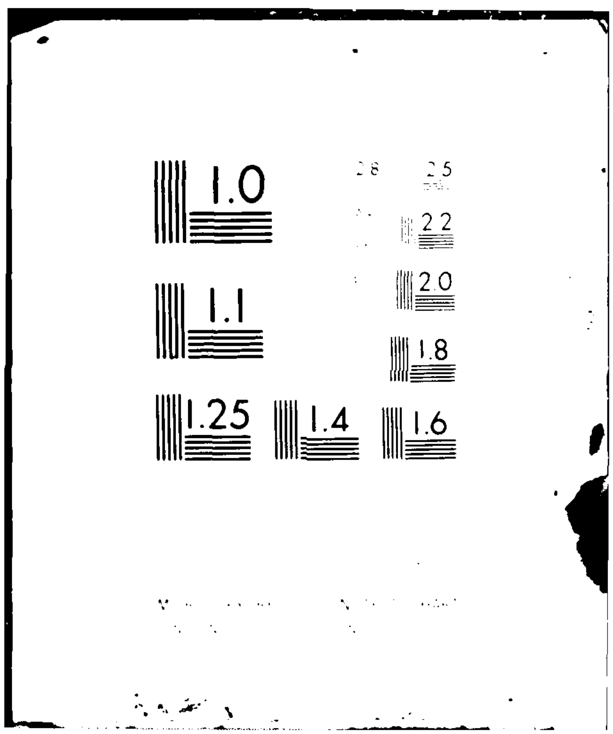
$$k_y = \frac{\phi(0,\tilde{\omega}_y) - \phi(0,0)}{\tilde{\omega}_y} \tag{162}$$

The program automatically makes these measurements and stores the corrections k_o, k_x, and k_y. The computed linear phase is then removed from each transform sample during the course of the measurements when the test image is in place. The samples which are available to the microprocessor from the sensor are in the form of the real and imaginary components of the transform. To remove linear phase, these samples must be converted to a magnitude and phase representation, the phase correction computed and applied and then the result converted back to real and imaginary format. The conversion is accomplished using the CORDIC algorithm. The total time involved per correction is about the same as the time required to compute two multiplies.

Once the parameters k_o, k_x and k_y have been measured the program request a sample average which is a power of two (2**S). The user then inputs the exponent Subsequent transform values will be sampled 2**S times per spatial frequency and averaged to improve the signal-to-noise ratio.

AD-A110 322 DEFT LABS INC EAST SYRACUSE NY F/G 20/1
ACOUSTO-OPTIC TECHNOLOGY FOR TOPOGRAPHIC FEATURE EXTRACTION AND--ETC(U)
MAR 81 A L MOYER DAAK70-79-C-0160
UNCLASSIFIED 0102-A002 ETL-0256 NL

AD-A
110322



1.0
2.8
2.5
2.2
2.0
1.1
1.8
1.25
1.4
1.6

The program in its present form is capable of computing all image moments from $m_{o,o}$ to $m_{8,8}$. To save time, if not all these are required, the user can specify a limit ≤ 8. Then only moments from $m_{o,o}$ to $m_{limit-1,limit-1}$ will be computed.

Next the program requests the user to position the test image and type "M". The program then computes the two-dimensional moments using equations (133), (134), (135) and (136). These moments are then normalized by $m_{o,o}$. It is easy to see that

$$m_{o,o} \geq m_{p,q} \tag{163}$$

for any p and q. Hence,

$$\frac{m_{p,q}}{m_{o,o}} \leq 1 \tag{164}$$

Then two moment invariants are computed. These are

$$X = a^2+b^2 \tag{165}$$

$$Y = a^2-b^2 \tag{166}$$

where $a^2 \pm b^2$ are defined by equations (82) and (83). The computed moments and X and Y are then printed out. The program then asks the user if he would like to continue. If he types a "Y" the program loops back and asks for a new sample average. If any other character is typed the program jumps to the monitor program and displays "*". A sample run of the program is shown in Figure 29. The notation 2**±(integer) means that the fixed point number to the left is to be multiplied by 2**±(integer). This is floating point using powers of two rather than ten. In the figure the case LIMIT = 3 is shown. If this parameter were larger then a larger table (up to 8 x 8) would be printed out.

```
POSITION UNIFORM IMAGE.  THEN TYPE C C
COMPUTES M(P,Q) MOMENTS FROM TRANSFORM

SAMPLE AVERAGE IS 2**3
LIMIT = 3
POSITION TEST IMAGE.  THEN TYPE M M
M(P,Q)
     0       1       2       3       4       5       6       7
0  +.9999  +.0055  +.1848
1  -.1204  -.0181  -.0707
2  +.1851  -.0468  +.0183

X = +.7109  2**-01  Y = +.6048  2**-04

CONTINUE?  Y
```

Figure 29 - Moments for Feature Extraction: Sample Run

For the particular implementation of the algorithm given in the program listed in Appendix B, the parameters n_e and n_o were set to be $n_e = 20$, $n_o = 40$ independent of p or q. This simplifies the computer program but results in run times which are longer than necessary because lower order moments can be accurately computed with much smaller values for n_e, n_o. The coefficients which were used in the program are listed in Tables 5 and 6. These coefficients can be utilized by a Deft sensor with bandwidth of 10 line pair along the ω_x-axis and ± 10 line pairs along the ω_y-axis. (See Section III. C.4 for a discussion.) Since the sensor which was available had a resolution of at least 20 line pairs along the ω_x-axis and ± 10 line pairs along the ω_y-axis, the available bandiwdth was sufficient for this application. Since n_e and n_o were too large for the lower power p and q, most of the coefficients in these cases are very small except for a few which are associated with transform samples near the Fourier origin. For example, consider $c_k(2)$. In this case only the first two or three coefficients are significant. Hence, lower order moments are insensitive to high spatial frequencies, as would

k	$c_k(0)$	$c_k(2)$	$c_k(4)$	$c_k(6)$
0	1.0000000	1.0000000	1.0000000	1.0000000
1	0.0	-0.5160232	-0.5864970	-0.6093051
3	0.0	0.0191080	0.1043204	0.1456567
5	0.0	-0.0041213	-0.0239249	-0.0501646
7	0.0	0.0014943	0.0088168	0.0202132
9	0.0	-0.0006940	-0.0041216	-0.0097799
11	0.0	0.0003693	0.0022002	0.0053087
13	0.0	-0.0002108	-0.0012579	-0.0030627
15	0.0	0.0001213	0.0007249	0.0017742
17	0.0	-0.0000636	-0.0003796	-0.0009316
19	0.0	0.0000198	0.0001184	0.0002910

Table 5 - Program Coefficients for p,q Even

k	$c_k(1)$	$c_k(3)$	$c_k(5)$	$c_k(7)$
1	-0.8109863	-0.8729509	-0.8872015	-0.8926364
3	0.0904811	0.2270004	0.2633326	0.2780638
5	-0.0328427	0.0826661	0.1262836	0.1463572
7	0.0169653	-0.0288055	-0.0660045	-0.0868873
9	-0.0104343	-0.0107975	-0.0356715	-0.0539768
11	0.0071316	0.0060242	0.0202588	0.0345133
13	-0.0052356	0.0042429	0.0124037	0.0227422
15	0.0040498	-0.0027277	-0.0081753	-0.0155041
17	-0.0032614	-0.0016602	-0.0056615	-0.0109329
19	0.0027129	0.0011839	0.0040415	0.0079358
21	-0.0023181	0.0009504	0.0029583	0.0058911
23	0.0020269	-0.0007069	-0.0022143	-0.0044456
25	-0.0018081	-0.0004914	-0.0016811	-0.0033913
27	0.0016420	0.0003742	0.0012820	0.0025989
29	-0.0015157	0.0003088	0.0009746	0.0019850
31	0.0014201	-0.0002323	-0.0007317	-0.0014942
33	-0.0013495	-0.0001532	-0.0005325	-0.0010885
35	0.0012995	0.0001024	0.0003619	0.0007408
37	-0.0012675	0.0000703	0.0002099	0.0004306
39	0.0012520	-0.0000271	-0.0000687	-0.0001412

Table 6 - Program Coefficients for p,q Odd

be expected. The coefficients in Tables 5 and 6 are stored in the computer program in Tables with labels C01, C23, C45 and C67. For example, C45 holds the coefficients for $c_k(4)$ and $c_k(5)$.

2. Method of Invariant Fourier Signatures

Since this algorithm uses the magnitude (or magnitude squared) of the Fourier transform, the phase correction described in the preceeding section is not required.

The overall flow diagram for the computer program is shown in Figure 30. A detailed flow diagram of projection computation is shown in Figure 31. The program contains a number of parameters which can be set to control the algorithm used. Projections can be computed which are the result of integration of the Fourier transform magnitude along radial lines, around (semi) circles or along spirals.

When the program is entered, the user is asked to specify a sample average as a power of two, S. Then all Fourier samples will be taken 2**S times and averages to improve signal-to-noise ratio. The user is then asked to specify circles or not circles. ("Circles" is a special case which requires different logic in the program.) If circles are chosen then the projection to be computed will contain 2**5 samples because the program always uses 32 radius values which are precomputed and stored in Table RVECT. The values of these radii satisfy the relationship given in equations (52) and (53). The projection results from integrating around the 2**5 semicircles.

If circles are not chosen the user can choose the parameter M in which case the projection will consist of 2**M samples. M should not be chosen larger than 6. M=5 is a good value to adequately sample the Fourier transform. If M=5 running time of the program is under one minute per image. The projection results from integrating the transform magnitude along 2**M radial lines or spirals.

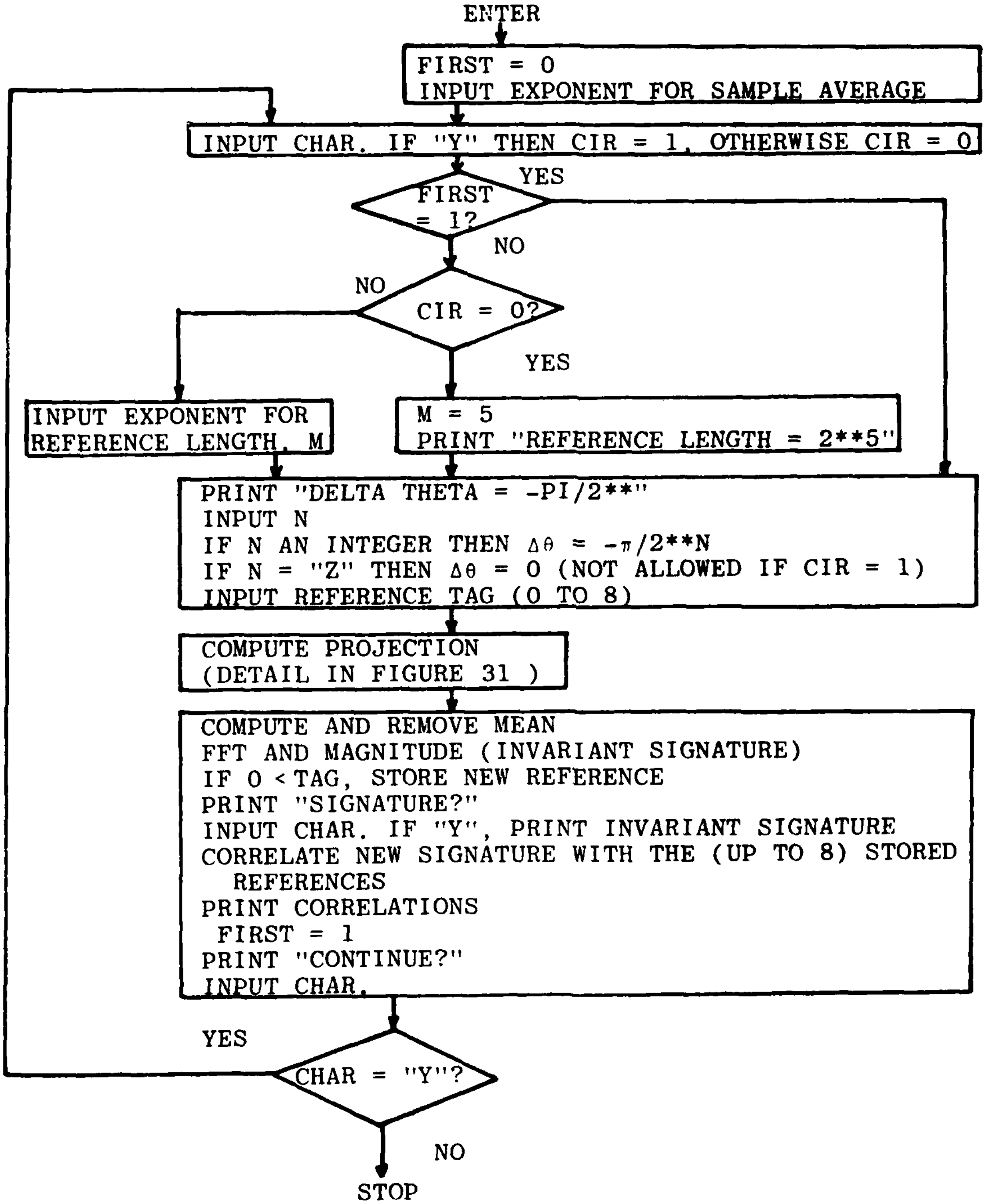


Figure 30 - Flow Diagram: Invariant Fourier Signatures

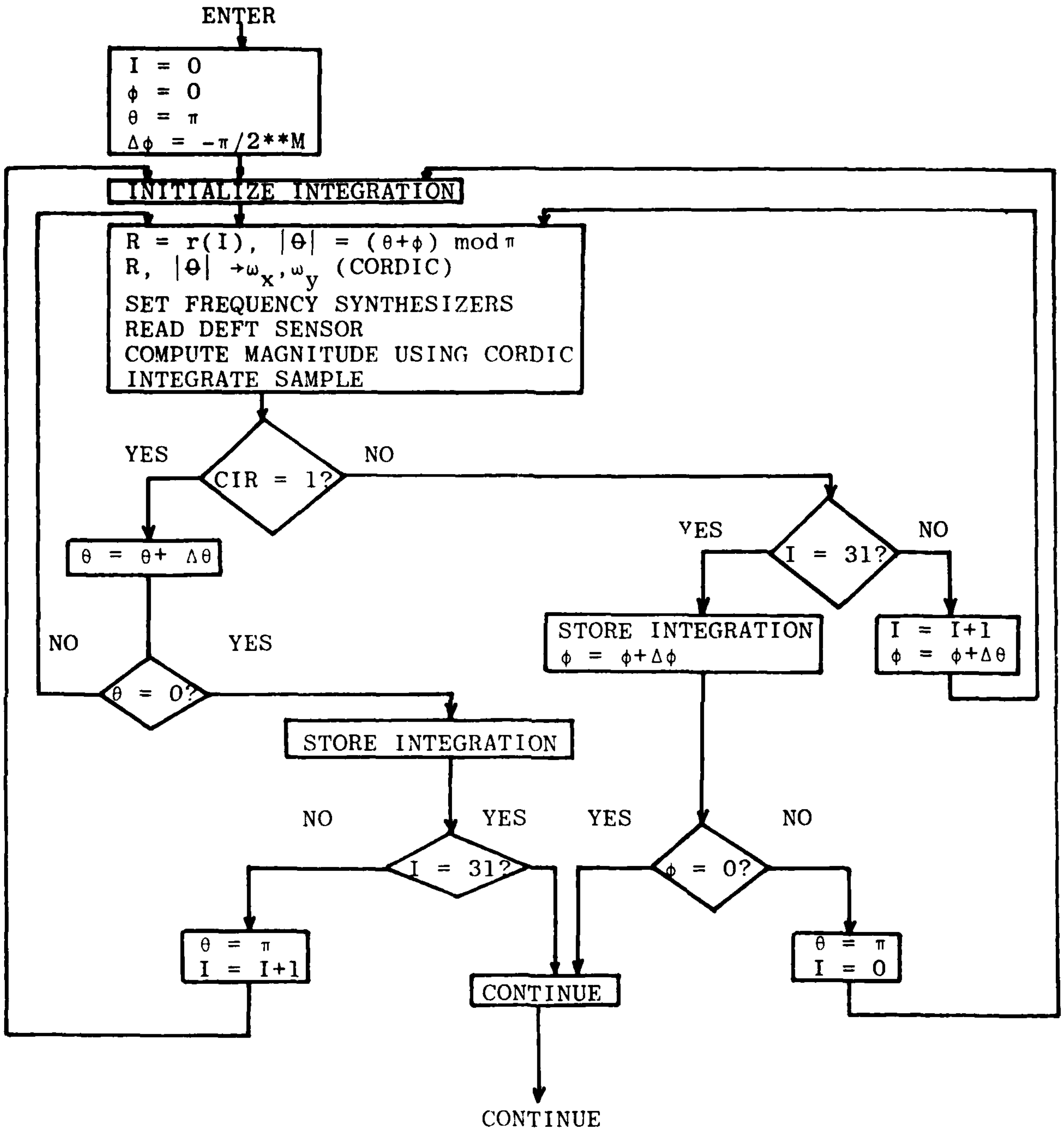


Figure 31 - Flow Diagram Detail: Projection Computation

The program then asks for the parameter $\Delta\theta$. $\Delta\theta$ has two functions depending on whether circles are chosen or not chosen. Consider "circles" first. In that case $\Delta\theta$ gives the angular spacing between samples along each semicircle. It also determines the number of samples which will be integrated along each semicircle. The finer the angle, the more the number of samples and the program running time. Again, $\Delta\theta = -\pi/2**5$ gives good sampling of the transform with a run time of under one minute. The polar coordinate of the kth sample along the i-th semicircle is given by

$$(r,\tilde{\theta}) = (r_i,(\pi+k\Delta\theta)_{\mathrm{mod}\pi}) \tag{167}$$

where r_i is the i-th radius. Angles are modπ because the transform is symmetric across the origin.

Now consider the case of "not circles". In that case $\Delta\theta$ gives the amount of skew for the spirals. Each spiral consists of 32 samples (one for each of the 32 radii.) The polar coordinates of the i-th sample along the k-th spiral is given by

$$(r,\tilde{\theta}) = (r_i, (\pi+i\Delta\theta +k\Delta\phi)_{\mathrm{mod}\ \pi}) \tag{168}$$

where

$$\Delta\theta = -\pi/2**M \tag{169}$$

If $\Delta\theta = 0$ then the contours of integration will be radial lines rather than spirals. To get $\Delta\theta = 0$, input "Z" when the program prints "DELTA THETA = -PI/2**". To get $\Delta\theta \neq 0$ enter the exponent instead. $\Delta\theta = 0$ will not be accepted by the program when "circles" since this would result in an infinite loop.

Next the program requests a reference tag. The user responds by entering an integer from 0 to 8. This is stored in variable TAG. If TAG>0 then after the invariant signature is computed it will be stored in a table in position TAG. Up to 8 references can be stored in the table. References can be replaced at any time by a new reference. The purpose of

these references is to provide a means of performing some simple feature recognition experiments. The user can set up a number of reference images consisting of distinct features. The program then computes the invariant signatures and stores them in the table. Then the user can use the system to view new images which might be translated, rotated or scaled versions of the references. In that case the user enters 0 for the reference tag. If TAG=0 the signature is then correlated against all the reference signatures. Let $\{r_i\}$ be the 2**M samples of a reference and $\{x_i\}$ be the samples of a new signature. Then the correlation is defined by

$$\text{correlation} = \frac{\sum_{i=1}^{2^m} r_i x_i}{\sqrt{\sum_{i=1}^{2^m} r_i^2}\sqrt{\sum_{i=1}^{2^m} x_i^2}} \tag{170}$$

This correlation is the basis for a simple decision processor. The signature is correlated with all reference signatures. These correlation coefficients are printed out by the program. The user can then observe these values and decide if the new image is a translated, rotated or scaled version of one of the references. A simple decision rule is to look for the largest correlation. If it is bigger than some threshold then the test image is considered to contain the feature corresponding to the reference signature at which maximum correlation occurred.

Returning to the discussion of the program, once the user enters the reference tag the program then computes the invariant signature. Sampling of the transform and integration of samples occurs in that portion of the program shown in Figure 31. There are two nested loops, one for each Fourier sample and one for each contour. The transform spatial frequency address is first computed in polar coordinates and then converted to rectangular coordinates using the CORDIC algorithm. The sensor is addressed and data read. The sensor output is converted to magnitude and phase using the CORDIC

algorithm. The magnitude is then integrated using the trapezoidal rule. At the end of each contour, the resulting integration is stored by being pushed onto the microprocessor stack. At the end of the last contour there will be 2**M samples in the stack.

Next, the program computes the mean of these samples and substracts the mean from each sample. This improves the accuracy of the FFT algorithm which is applied to these samples. The algorithm uses fixed point arithmetic with a block floating point scaling scheme. If the mean is not removed the FFT output will have a large component in the zero frequency bin. This leads to poor scaling for the remainder of the FFT bins which are, typically, much smaller in magnitude. Removing the mean corrects for this problem.

The FFT output samples are then converted to magnitude and phase representation using the CORDIC algorithm. The resulting 2**M-vector of real numbers is the desired invariant signature.

The program then prints "SIGNATURE?". If the user enters "Y" the invariant signature is printed out. If instead "N" is entered, this is skipped. The program then stores the signature of TAG>0. It then correlates with all reference signatures and prints out the correlation coefficients.

The program then prints "CONTINUE?" Entering a "Y" causes the program to ask for new parameters for another run. Entering an "N" causes a jump to the monitor and a "*" is displayed.

A sample run of the program is shown in Figure 32. The algorithm used in this run was 2**5 radial lines. There was no sample averaging. The first three correlation tables represent three references. The fourth table gives the correlation of a misaligned feature with the three references. In the case shown, correlation with reference 1 was largest. The test object was reference 1 scaled by a factor of 0.9 in size.

```
FEATURE RECOGNITION
DEFT/PROJECTION/FFT

SAMPLE AVERAGE IS 2**0
CIRCLES?N
REFERENCE LENGTH=2**5
DELTA THETA=PI/2**Z
REFERENCE TAG=3
SIGNATURE? N

CORRELATION
1) 0.0000  2) 0.0000  3) 0.9999  4) 0.0000
5) 0.0000  6) 0.0000  7) 0.0000  8) 0.0000

CONTINUE? Y
CIRCLES?N
DELTA THETA=-PI/2**Z
REFERENCE TAG=2
SIGNATURE? N

CORRELATION
1) 0.0000  2) 0.9999  3) 0.7623  4) 0.0000
5) 0.0000  6) 0.0000  7) 0.0000  8) 0.0000
CONTINUE? Y
CIRCLES?N
DELTA THETA=-PI/2**Z
REFERENCE TAG=1
SIGNATURE? N

CORRELATION
1) 0.9999  2) 0.8576  3) 0.8727  4) 0.0000
5) 0.0000  6) 0.0000  7) 0.0000  8) 0.0000
CONTINUE?

          Y
CIRCLES?N
DELTA THETA=-PI/2**Z
REFERENCE TAG=0
SIGNATURE? N

CORRELATION
1) 0.9795  2) 0.9364  3) 0.8578  4) 0.0000
5) 0.0000  6) 0.0000  7) 0.0000  8) 0.0000
```

Figure 32 - Invariant Fourier Signatures: Sample Run

C. Feature Extraction Experiments

The two computer programs which are described in the previous section were written for the purpose of performing some preliminary feature extraction experiments utilizing the algorithms developed in Section III and the hardware system described in this section. It was not the intent of these experiments to show detection of real features in aerial imagery. Rather, the purpose was to verify feasibility of both the algorithms and the sensor technology to detect features from a small set of controlled text patterns. In this way, directions for further improvements in both algorithms and sensors could be determined.

It was originally intended to perform experiments using the method of invariant Fourier signatures (IFS) and the method of invariant moment signatures (IMS). However, the program which computes moments from the Fourier transform gave poor results. The reasons for this were identified. It was determined that the program computes moments with large errors because of two shortcomings of the Deft sensors which were used in the experiment. Since being identified, steps are being taken to correct this problem in future Deft sensors. More will be said about this in Section V. Briefly, however, the two sensor shortcomings were the following:

1. Because of the current collecting metal bars used in the present sensor design strong reflections will occur in the surface acoustic wave propagating orthogonal to these bars. (Refer to Figure 2.) This is because the periodic structure

of these bars reinforces the small reflections which occur at each bar. The effect of these reflections on the Fourier transform is to multiply the transform by a weight function of, say W_x. That is, the function which is available is $\tilde{F}$ where

$$\tilde{F}\ (\omega_x,\omega_y) = W(\omega_x)\ F(\omega_x,\omega_y) \tag{171}$$

Visually, $|\tilde{F}|$ is a rippled version of $|F|$. This can be seen in any of the transform plots which are given in Section V. One problem caused by these ripples is that it is not possible to accurately determine the origin of the Fourier transform. The origin will always be located at the peak of the zero frequency sinc function. However, because of the ripples on this peak, the true maximum cannot be determined. It can be approximately determined by plotting the peak and visually determining the center.

The second problem with ripples is that the transform will contain errors because of $W(w_x)$. Refer to Tables 5 and 6 Notice that for the lower order moments the coefficients C_k are largest for small indices k. This implies that components of the transform nearest the origin will be most heavily weighted in computing these moments. Refer, for example to $C_k(2)$. It can be seen that the computed moment will depend primarily on the value of the transform at the origin minus the weighted value of the transform at $\omega = \pi/4$ (See Figure 25) If the peak of a ripple occurs at $\omega = 0$ and the trough of the ripple occurs at $\omega = \pi/4$ then the computed moment will be larger than the correct value. Other situations leading to other types of errors can be envisioned.

By examining the transform plots in Section V it is obvious that the large ripples shown there would lead to large errors in computing image moments. This proved to be the case in attempts to perform experiments. For example, in some cases moments $m_{p,q}$ where both p and q were even were computed to have negative values. However, these moments must always be positive.

The solution to this problem is to redesign the Deft sensor to eliminate the reflections. This is presently being done as will be detailed in the next section.

2. The second shortcoming arises from other inaccuracies in computing the Fourier transform. These will be disccused in Section V. As mentioned, when viewing a completely uniform image, the Deft sensor output contains a linear phase term which must be removed to compute moments. However, because the transform is computed with some error, the assumption of linear phase in only an approximation. In attempting to use the computer program it was determined that the assumption of linear phase was only approximately true. Hence, the undesirable phase function can only be approximately corrected for. This is not a serious problem in the computation of low-order moments since they depend primarily on transform samples near the Fourier origin where on accurate phase correction can be made. However, the accurate computation of moments above the first few would require much better phase corrections. It is difficult to presently make such corrections for two reasons. First, the uniform, white image used as a text pattern has a transform whose magnitude is large over only a limited region of Fourier

space. It is only possible to measure phase in this region with this test pattern. Accuract phase measurement over the entire transform would require a large number of accurately positioned test patterns. Secondly, the Fourier transform output of the sensor is only an approximately linear function of light intensity. Hence, phase corrections measured at one intensity may be in error when the image to be analyzed has another average intensity.

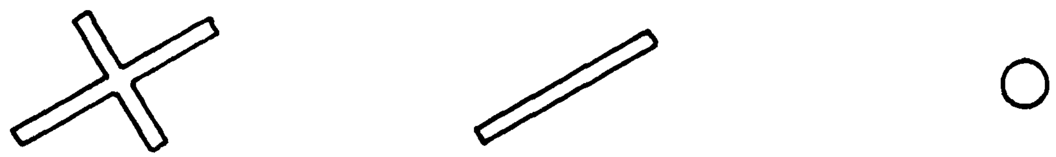
Efforts are underway to improve the accuracy of the computed transform. This should lead to more predictable and accurate phase measurement and correction. These efforts are discussed in the next section.

An alternative solution would be to compute the moments of the image autocorrelation function rather than of the image. That is, use the magnitude of the Fourier transform rather than the real and imaginary parts. This solution circumvents the phase problem but may be undesirable in that some image information is lost.

The remainder of this section is a discussion of the experiments conducted using the method of invariant Fourier signatures. In Part B.2 of this section the flexibility of the computer program was described . The algorithm developed in Section III can be configured in a number of ways using constants entered into the program. These constants control the sample averaging of the data, the angle ϕ of the projection $P\phi$ and the number projections computed. In application, these parameters can be chosen to optimize the invariant signatures which are computed given a class of features to be detected. However, because of the scope of this program, only a limited

number of algorithm configurations were utilized. To be specific, two forms of the algorithm were used in all tests. These are integration along 32 radial lines and integration around 32 circles. No sample averaging was used in the experiments after it was determined that averaging did not affect the results. This was desirable since averaging increases program run time.

In all experiments, three reference objects were used. These references are shown in Figure 32. The images consisted of a dark background with white features.



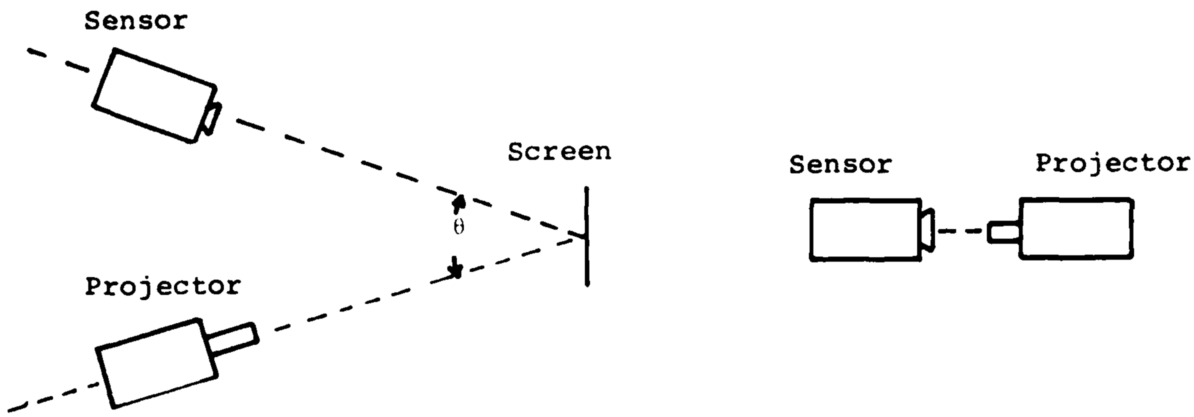
a. "Crossroads" b. "Road" c. "Storage Tank"

Figure 33 Test Reference Features

These features are simple geometric objects. However, they resemble some important realistic features and were chosen on this basis. These features are a crossroads, a road and a compact, round structure such as a storage tank. The reference objects are meant to only be idealizations of these features. In particular, the images contain no background noise. A Viewlex projector with a rotating barrel was used to project the images which were in the form of slides. The 60 degree tilt of the images is a result of the Viewlex. The rotating barrel had a stop at which the slides were tilted 60 degrees. Rotation of images was then measured with respect to this stop. This 60

degree tilt has no significance.

The experimental set-ups which were used shown in Figure 33.



a. Indirect Projection b. Direct Projection

Figure 34 Experimental Set-Ups

In some experiments the image was projected on a screen and the sensor viewed the projected image. The angle θ between projector and sensor was minimized to prevent image distortion. This set-up was used in the experiments where image scale was to be varied. This was accomplished by moving the projector along the dashed line and refocusing. The size of the image was measured on the screen using a ruler. In the other experiments the lens on the Deft sensor module was removed and the projector was placed in line with the sensor. The image was then focused directly, at close range, onto the Deft sensor. To do this it was necessary to stop down the projector lens using the f-stop adjustment. The module was mounted on a triangular rail so that it could be accurately translated at right angles to the dashed line in Figure

33b. This was used in the experiments where the feature to be detected was a translated version of the reference. The lens barrel was rotated in experiments where the feature to be detected was a rotated version of the reference.

During the period of time when experiments were conducted, two Deft sensors were available for test. These will be labeled sensor #1 and #2. The experiments will now be described. In all experiments the first step was to compute the three reference signatures using the images of Figure 32 in their reference positions. Then the system was presented with the same images but either rotated, scaled or translated or a combination of these. The signature was then computed and correlated against the three reference signatures. The reference at which the highest correlation occured was then considered to be the detected feature. If this was the correct feature then the system "passed". If not, the system "failed". The specific experiments conducted were the following:

Image Rotation

Reference signatures were computed. The system was then presented with the same images but with the projector barrel rotated from 0 to 25 degrees in 5 degree steps.

Image Translation

Reference signatures were computed. The system was then presented with the same images but the sensor module was translated from 0 to 8 mm in 1 mm steps. The width of the active sensor area is 12.7 mm. Generally, translations of greater than 8 mm resulted in a significant part of some of the images falling outside the sensor active area.

Image Scale

Reference signatures were computed. The projector was then moved closer to the screen and refocused so that a dimension on one of the references was reduced by a factor of k. This was repeated for k = {.9, .8, .7, .6, .5}. The system was presented with the same three images at each position of the projector.

Image Translation with 5 Degree Offset

Reference signatures were computed. The projector lens barrel was then rotated 5 degrees. The system was then presented with the same images but with the sensor module translated from 0 to 8 mm in 1 mm steps.

Image Translation with 10 Degree Offset

Same as above but with a 10 degree offset after the reference signatures have been computed.

Image Translation with 15 Degree Offset

Same as above but with a 15 degree offset.

Image Rotation with k = .7 Scale Offset

Reference signatures were computed. The projector was then moved toward the screen until all image dimensions were reduced by a factor of 0.7. The system was then presented with the same images but with the projector barrel rotated from 0 to 25 degrees in 5 degree steps.

The results of these experiments are given in Tables 7, 8, 9, and 10. These tables give the number of passes and fails as a function of sensor, algorithm, experiment and reference pattern. Both algorithms performed approximately equally.

$$\text{success rate} = \frac{\text{number of passes}}{\text{number of passes + number of failures}} \times 100\% \tag{172}$$

If only the experiments of rotation, translation and scale are considered then the success rate for all experiments and patterns are as follows:

Sensor #1, 32 radial lines: 81.7% success rate

Sensor #1, 32 circles: 81.7% success rate

Sensor #2, 32 radial lines: 86.7% success rate

Sensor #2, 32 circles: 80.0% success rate

As can be noted, both sensor performed approximately equally.

	Reference Pattern							
	1		2		3		Total	
Experiment	P	F	P	F	P	F	P	F
Rotation	4	2	5	1	6	0	15	3
Translation	7	2	9	0	6	3	22	5
Scale	5	0	5	0	2	3	12	3
Translation with 5 degree offset	2	7	9	0	8	1	19	8
Rotation with k=0.7 scale offset	4	2	6	0	4	2	14	4
Total: all experiments	22	13	34	1	26	9	82	23

Table 7 - Feature Extraction Experimental Results: Sensor #1, 32 Radial Lines

	Reference Pattern							
	1		2		3		Total	
Experiment	P	F	P	F	P	F	P	F
Rotation	6	0	6	0	6	0	18	0
Translation	7	2	6	3	7	2	20	7
Scale	3	2	5	0	3	2	11	4
Translation with 5 degree offset	5	4	6	3	8	1	19	8
Rotation with k=0.7 scale offset	6	0	5	1	3	3	14	4
Total: all experiments	27	8	28	7	27	8	82	23

Table 8 - Feature Extraction Experimental Results: Sensor #1, 32 Circles

	Reference Pattern							
	1		2		3		Total	
Experiment	P	F	P	F	P	F	P	F
Rotation	6	0	6	0	6	0	18	0
Translation	9	0	9	0	8	1	26	1
Scale	3	2	5	0	0	5	8	7
Total: all experiments	18	2	20	0	14	6	52	8

Table 9 - Feature Extraction Experimental Results:
Sensor #2, 32 Radial Lines

	Reference Pattern							
	1		2		2		Total	
Experiment	P	F	P	F	P	F	P	F
Rotation	6	0	6	0	6	0	18	0
Translation	9	0	2	7	9	0	20	7
Scale	3	2	5	0	2	3	10	5
Total: all experiments	18	2	13	7	17	3	48	12

Table 10 - Feature Extraction Experimental Results:
Sensor #2, 32 Circles

It was observed during the experiments that failure occurred more frequently where there was either a large translation, rotation or scale offset between references and test patterns. It has been determined that this is because of certain errors in the Deft sensor output. More will be said about these errors in Section V.

Some additional, more challenging experiments were performed using sensor #1. These experiments involved offsets of either translation and rotation or rotation and scale between reference and test images. The results of these experiments are also tabulated in Tables 7 and 8. For these experiments only, the success rates were the following:

Sensor #1, 32 radial lines: 73% success rate

Sensor #1, 32 circles: 73% success rate

as can be seen, the additional misalignment reduced the success rate somewhat. This is a consequence of the same Deft sensor errors alluded to above.

These experiments verify the premise of invariance to translation, rotation and scale. However, the success rates must be described as modest. Part of the difficulty can be traced to sensor performance, but another part may result form the algorithms which were used. One of the features of an ideal prefilter mentioned in Section III is that the prefilter increases the between-class seperation of the feature classes. For the present experiments between-class seperation is measured by the crosscorrelation of signatures of each of the reference patterns.

Table 11 Reference Pattern Correlations

Reference Pattern		1	2	3
Reference Pattern	1	0.9754	----	----
	2	0.9225	0.9961	----
	3	0.8151	0.7508	0.9749

a. Sensor #1, 32 Radial lines

Reference Pattern		1	2	3
Reference Pattern	1	0.9991	----	----
	2	0.9877	0.9989	----
	3	0.9495	0.9751	0.9970

b. Sensor #1, 32 Circles

Reference Pattern		1	2	3
Reference Pattern	1	0.9943	----	----
	2	0.8551	0.9982	----
	3.	0.7819	0.9218	0.9696

c. Sensor #2, 32 Radial Lines

Reference Pattern		1	2	3
Reference Pattern	1	0.9994	----	----
	2	0.9891	0.9996	----
	3	0.9522	0.9827	0.9990

d. Sensor #2, 32 Circles

The crosscorrelations and auto correlations of the reference signatures are given in Table 11 for sensors #1 and #2 and for both algorithms. The entries represent the average of all values which were obtained in all the experiments tabulated in Tables 7 through 10. Notice that the crosscorrelation terms were not significantly smaller than the autocorrelation terms. This was particularly the case for the "circles" algorithm. Hence, between-class separation was not as large as might be desired. This drawback cannot be blamed on the sensor. Rather it is a consequence of the algorithms used. It is the feeling of this Author that the requirement of the second transform (FFT) tends to smooth the data in such a way that there are not large differences between the signatures of different reference patterns. It appears that by designing invariance into these algorithms and thereby decreasing the in-class seperation, the between-class seperation also decreases.

The experiments which were conducted used noise-free, idealized features. Since the performance reported here is only modestly successful it is evident that the present Deft sensor is not capable of detecting features in realistic aerial images making use of the algorithms developed in this report. It is possible that present sensors could be useful in more limited applications such as the detection of man-made vs. natural features. The utility of the Fourier transform in this application has been verified by Lendaris and Stanley.[20] Their algorithms do not require a second transform and exhibit greater between-class seperation. However, they are not invariant to translation rotation and scale.

In order to take advantage of the more powerful algorithms developed in this report, a sensor which faithfully produces the Fourier transform is required. The basic requirements of the sensor output are the following:

1. The magnitude of the sensor transform should be invariant to feature translation.

2. If the feature rotates then the magnitude of the transform must also rotate.

3. In order to include filtering against additive noise the sensor must be linear. That is, if image I_1 has transform F_1 and image I_2 has transform F_2 then the combined image I_1+I_2 has transform F_1+F_2.

V. ACOUSTO-OPTIC SENSOR CAPABILITIES: PRESENT AND PROJECTED

A. Introduction

This section deals with the present capabilities and limitations of the A-O devices considered in this study. Capabilities and limitations are considered in the context of feature extraction. Also considered are projected improvements and the probability of success.

Since the Deft sensor is presently receiving active development support by Deft Laboratories Inc., more detailed information can be provided on this sensor than the others which were discussed in Chapter 2. Recent and present Deft sensor development has been funded by NASA and by internal Deft Laboratories Inc. funds. The level of NASA funding for Deft sensor development in the last year is $130,000.

This chapter begins with the elastobirefringent light valve. The Deft sensor is covered in the following section. The Thomson - CSF sensor can be thought of as a specialized Deft sensor so that it will not receive a separate discussion.

B. Elastobirefringent Light Valve

This sensor is not presently receiving development support. Devices which have been developed to date should be considered experimental. In order to apply this sensor to feature extraction present device limitations must be overcome and manufacturing procedures developed. Two years of additional development are required for this sensor to be compatible in performance with presently available Deft sensors.

Some aspects of current Deft sensor development could be utilized in improving the performance of the bulk acoustic wave (BAW) light valve. For example, a segmented transducer is currently under development which will produce wider transducer bandwidths. This new design could be used to increase the bandwidth of the BAW device also.

Other developments would have to be undertaken however, which are specifically required for the BAW sensor. In order

to achieve wider transducer bandwidths, small transducers with higher center frequencies are required. In the experimental BAW sensors, transducers were glued onto the quartz cell. The smaller transducers would be thinner and more brittle so that it would be necessary to sputter transducers directly onto the quartz cell.

Standing waves in the sensor have been mentioned as a current problem. These waves result from the reflection of the BAW from the cell boundaries. A number of techniques could be used to absorb the unwanted acoustic energy. For example, a larger cell could be used with only part of the cell serving as the active sensor volumn. The remainder of the cell would then be used as an absorbing volumn for the BAW. The edges of the cell could be joined with an absorbing material. The edges might also be sandblasted or made jagged to trap the acoustic energy. Another technique would be to drill small, random holes in the non-active volumn of the cell to scatter the BAW. This could be done using a laser.

Present sensors exhibit a number of other problems which are discussed in Section II. Two to three years are required to produce a device suitable for feature extraction application. Since this work is developmental, the probability of success is perhaps 0.5

C. Deft Sensor

The experimental results of Section IV indicate that presently available Deft sensors are capable of limited feature recognition. These sensors are being improved through a program of active support. Hence, the probability is good that improved sensors will be available which will have a realistic and significant feature recognition capability. This section will detail the current limitations and the steps being taken to correct these limitations.

The magnitude of the Fourier transform of the three reference features shown in Figure 32 as computed by a Deft

sensor are shown in Figures 34, 35 and 36. Figures 34 and 35 show the transform of two reference features in two different angular orientations. It can be seen that the transforms are (approximately) rotated versions of each other. Figure 36 shows the transform of a circle which was positioned at two locations on the sensor. These transforms are also (approximately) the same. The properties of translation invariance and transform rotatation are required for all the algorithms developed in this report. Hence, the Deft sensor is potentially very useful for feature extraction. However, these properties are only approximately true for present sensors. Deviations in these properties as well as some other limitations restrict the usefullness of present Deft sensors in feature extraction. Limitations exist in the following categories:

1. spatial bandwidth
2. acoustic reflections
3. CdS uniformity
4. transform phase accuracy
5. output signal level

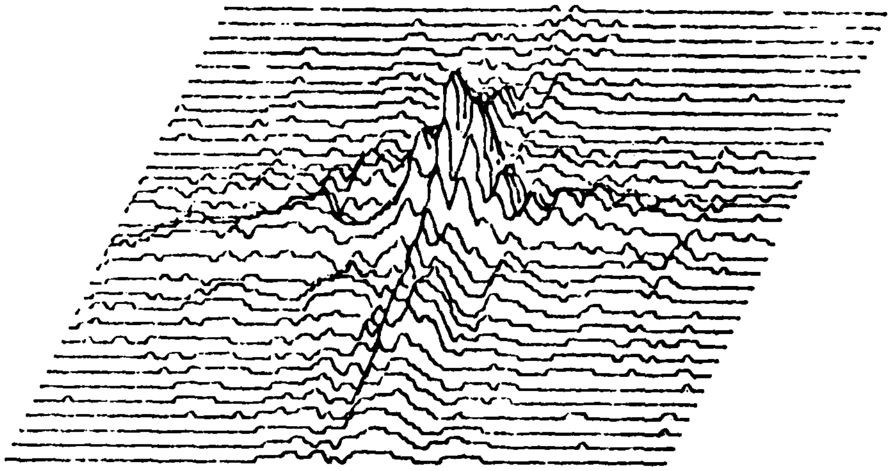
These categories will now be discussed one-by-one.

Present Deft sensors have spatial frequency bandwidths of ± 10 line pairs along one transform axis and +20 line pairs along the other axis. A larger bandwidth is desirable for the following three reasons:

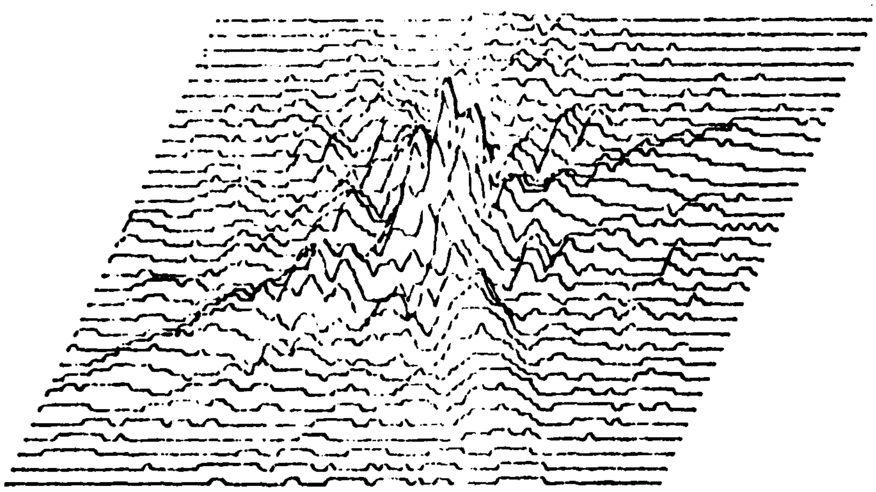
1. To distinguish between features which are only subtly different from each other, high frequency information is required since the transforms may only differ in high frequency content.

2. Since the Fourier transform scales as the inverse of the feature scale, small features have large transforms. Large spatial bandwidths are needed to "see" large transforms. If bandwidth is limited then the image must be magnified. This requires breaking the photograph into a large number of views thereby increasing processing time.

3. If the scene contains "noise" which corrupts portions

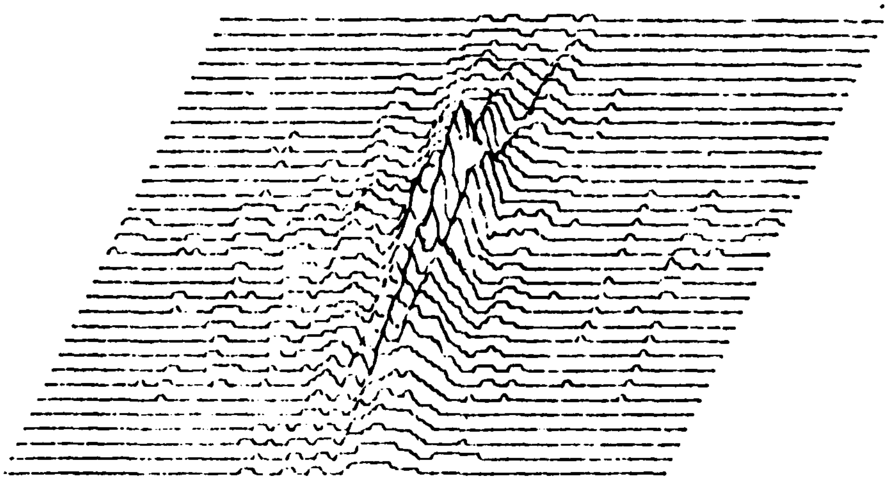


a. θ = 0 degrees

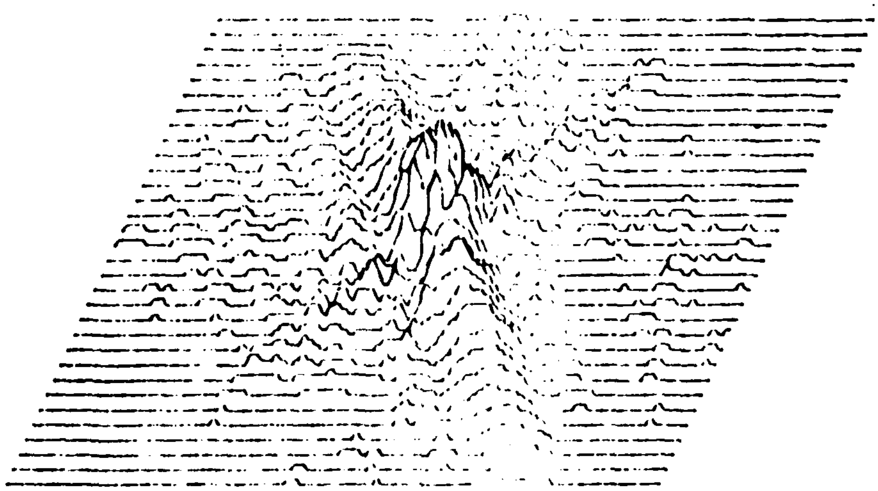


b. θ = 25 degrees

Figure 35 - Deft Transform of "Crossroads" Feature

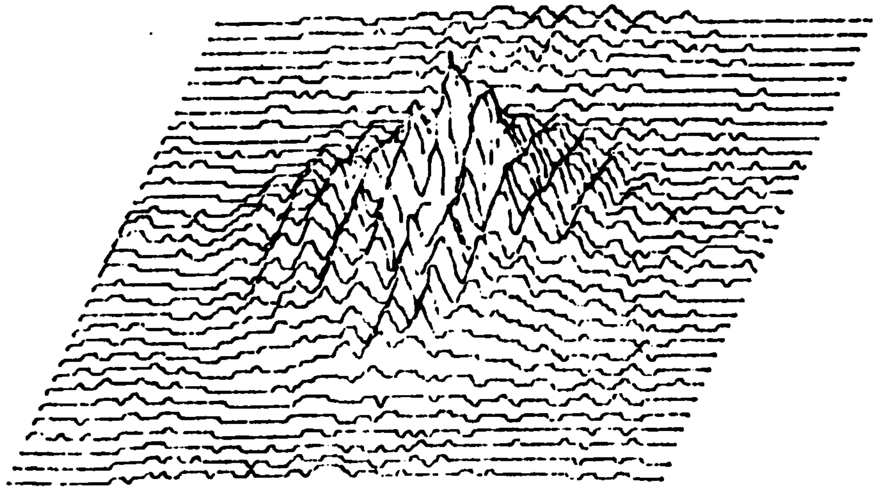


a. θ = 0 degrees

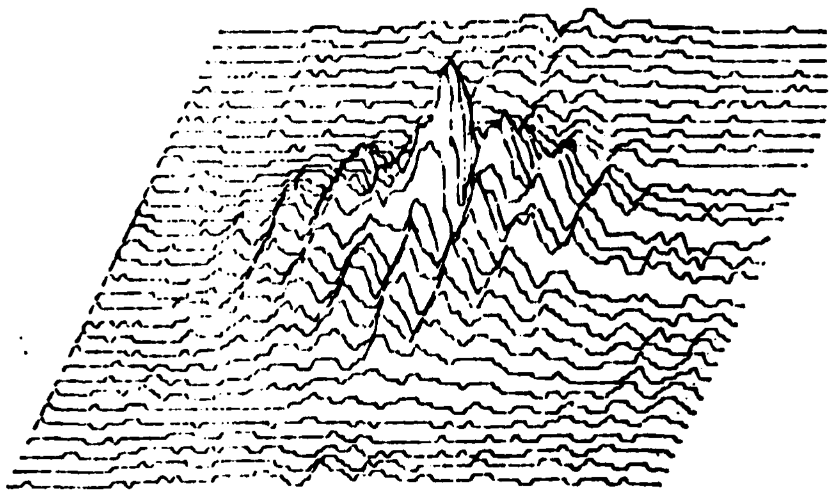


b. θ = 25 degrees

Figure 36 - Deft Transform of "Road" Feature



a. Circle at Center of Sensor



b. Circle at Edge of Sensor

Figure 37 - Deft Transform of Small Circle Feature

of the transform, a larger bandwidth may be required in order to have sufficient, useable transform components.

Deft sensors which will have larger bandwidths are under development. A sensor will be available by June 1981 which will have spatial frequency bandwidths of ± 25 line pairs along one transform axis and +28 line pairs along the other axis. This sensor will be able to "see" approximately four times the number of spatial frequency components as do current devices. This will be termed a "medium resolution" sensor. A "high resolution" sensor is also planned which should be available by January 1982. This sensor will have a fourfold increase in resolvable spatial frequencies over the medium resolution sensor.

In order to improve resolution, the medium resolution sensor will be fabricated on 41.5^{o} rotated z-cut $LiNO_3$. The transducer center frequencies will also be increased from the present 35-40 MHz range to about 60 MHz. Increasing transducer center frequencies will result in a greater absolute bandwidth from a given percentage bandwidth. This is an attractive approach up to about 60MHz. Beyond this frequency, the mechanical loading of the metal grid pattern on the substrate leads to significant and undesirable damping of the SAW.

There are three advantages to using the new, 41.5^{o} rotated z-cut $LiNO_3$. First, the coupling constants are larger than for the old cut. This means that more acoustic energy can be introduced into the sensor resulting in a larger output signal. Secondly, the spurious response of the sensor to a uniform image is smaller and further away from the main response peak than was the case for the old cut. A discussion of spurious acoustic modes in Deft sensors is found in Reference 31. If the sensor bandwidth were increased then the spurious would be within the bandwidth. Hence, it is important that the spurious is as small as possible. The third advantage is that with the rotated cut of $LiNO_3$ the number of transducer finger pairs required for optimum energy coupling decreased from 8 to 6 in

one direction and from 8 to 4.5 in the other direction. Fewer finger pairs mean a larger percentage bandwidth.

The high resolution sensor will include these improvements. This sensor will also include an improved transducer design. The new transducers will be segmented in order to reduce their capacitance. Reduced capacitance allows wider transducer bandwidths. The resulting bandwidth leads to a significantly greater utility of Deft sensors in feature extraction.

One of the limitations of current Deft sensors is that these devices exhibit acoustic reflections from the regularly-spaced metal pick-up fingers. These reflections lead to a scalloping of the transform magnitude. This is evident in Figures 34, 35 and 36. The ripples which can be seen make feature extraction more difficult. For example, in Figure 37 by moving the circle from the center to the edge of the sensor the shape and the number of ripples changes. The feature has remained the same but the transform is really not invariant to translation which is a requirement for the preprocessor algorithms. Now refer to Figures 35 and 36. Notice that when the feature is in two different angular orientations the ripples may or may not appear on the "arms" of the transform. Hence, the transform is really not just rotated when the image rotates. There is also a shape change which is not desired. A third problem with these ripples occurs in the computation of image moments. The lower order moments are sensitive to the partial derivatives of the transform at the origin. Because of the ripples, these derivatives are drastically altered. In addition, they also become a function of feature position. In summary, ripples caused by reflections are very undesirable.

The medium resolution sensor will incorporate a new metal finger pattern spacing which will eliminate the reflections. This new spacing causes reflections from metal lines to add distructively. An important advantage of this spacing is that critical metal-to-metal dimensions are not reduced. In

general, device yield goes down as metal-to-metal dimensions are reduced because it is then more likely that contaminates will cause shorts across the metal lines.

A problem with current Deft sensors is that the CdS squares on the sensor do not conduct uniformly. Measurements have been made showing up to 2:1 variation in light conductivity over a sensor. This variation weights the image function. In the transform domain the image transform is convolved with the transform of the weight function. This tends to broaden spectral peaks and reduce or eliminate sidelobes. We typically observe peaks which are twice as wide as theory predicts. CdS variation leads to another problem. The computed transform may not be invariant to translation or rotate as the image rotates. This is evident in Figures 34, 35 and 36.

It has been determined that film variation is a result of the method used to deposit CdS on the substrate. The material, in the form of a gas, is passed over the substrate in a furnace. The substrate is almost as large as the furnace tube so that flow rates near the substrate edges are faster than at the middle. We presently have on order a new furnace tube with a larger inner diameter. Uniform flow rates should give uniform CdS film. An additional solution is to reduce the flow rate into the furnace in order to reduce turbulance. The success of these changes will be measured when the new equipment is in-place.

Another problem which was discussed in Section IV is the inaccuracy of the transform phase computed by the sensor. This is important in the method of moments. Part of the phase error must come from the same sources which produce errors in the transform magnitude. However, the sensor contains a phase term which is a function of the electronics (amplifiers, filters) and not the transform. These electronics were not originally designed to provide a linear phase response. Some recent measurements have been made which show that, indeed, the phase is nonlinear. Since these filters must be redesigned to

accomodate the new medium resolution sensor, they will be redesigned to exhibit approximate linear phase response.

The final item to be considered is output signal level. Present output signals are of the order of a microamp. In order to amplify this signal and achieve a good signal-to-noise ratio at the amplifier output, a rather narrow amplifier bandwidth is required. This narrow bandwidth affects the rate at which the sensor drive frequencies can be changed. If signal-to-noise ratio was not a problem then drive frequencies could be stepped every two microseconds which is the time required to propagate the SAW across the sensor. This would allow 5×10^5 Fourier components to be addressed every second. But because of the need to narrow the output bandwidth, this rate is reduced to about 10^3 components per second. If the sensor output signal could be increased by a factor of 10 then 10^4 components per second would be possible.

The new medium resolution sensor will have an output signal approximately eight times as large as current sensors. This is because the 41.5^o rotated z-cut $LiNO_3$ results in better coupling of energy into the SAW.

All the factors which have been identified as limiting the performance of the Deft sensor in feature extraction applications have been listed and improvements under development discussed. In a number of cases, expected improvements can be stated with high probability of success. In other cases, improvements must be determined after new devices are fabricated and tested. The medium resolution sensor will be available by June 1981. The high resolution sensor will be available by January 1982. These sensors should allow significant improvements in feature extraction capability. It is suggested that these new sensors be examined for their applicability to feature extraction at the time they they are available. This could be done with little cost by simply repeating the feature extraction experiments using the presently available computer programs and the new design sensors.

VI. SUMMARY AND CONCLUSIONS

The objective of this program was to develop, analyze and evaluate theoretical concepts and strategies for topographic feature extraction and image analysis using accousto-optic (A-O) technology. To provide a frame of reference, a general feature extraction system model was developed in Section I. The most computation-intensive portion of the feature extraction process is the prefilter function. The purpose of this function is to reduce the large information content of the image to a much smaller set of values which can then be input to the decision processor. An important conclusion of this study was that A-O devices are potentially capable of implementing the prefilter function very efficiently. The input/output function of these devices is defined by equation (1). The device input is an image which is then transformed, using equation (1), into an electrical signal which can be input to a digital decision processor. The A-O devices have a number of desirable properties in this application. First of all, the device input/output format is ideal for the application. Secondly, lasers or precise optical alignments are not required. Third, these devices are rugged and potentially inexpensive. Finally, the function defined by equation (1) is central to a number of promising prefilter algorithms.

Transform-based prefilter algorithms were examined in Section III. Algorithms were developed there which are invariant to feature translation, rotation and scale. This invariance is highly desirable since it reduces the number of distinct feature signatures which must be processed by the decision processor. These algorithms require, as input, either the image two-dimensional Fourier transform or the image two-dimensional moments. Either of these functions can be computed efficiently using A-O devices.

Some preliminary experiments were conducted using the Fourier-based algorithms, test images and an A-O device which

was a Deft sensor. These experiments verified the invariance properties of the algorithms. The microcomputer-based Deft system was able to distinguish between three test patterns which were presented to the system in arbitrary orientation and scale. The success rate was 80%.

In spite of these promising results, present Deft sensors are not capable of distinguishing realistic features in aerial photographs. Present sensor limitations are identified and discussed in Section V. New Deft sensors are presently under development which will significantly improve the capability of this sensor in feature extraction applications. The expected improvements in a number of parameters can be predicted fairly accurately. However, other parameters must be measured after the new devices are available. It is suggested that the best way of determining the applicability of new sensors to feature extraction is to rerun the experiments reported here using the already written computer programs but using the new sensors. Since the microprocessor-based Deft system is already in-place and the computer programs written, these investigations could be accomplished with only a few man-weeks of effort.

VII. REFERENCES

1. M. D. Levine, "Feature Extraction: A Survey," Proc. IEEE, Vol. 57, No. 8, pp. 1391-1407, Aug. 1969.

2. P. Kornreich, S. Kowel, A. Mahapatra and M. Mehter, "Monolithic Two-Dimensional Fourier Transformer," 1978 Int. Optical Computing Conference, pp. 49-54.

3. S. Kowel, P. Kornreich, A. Mahapatra, T. Szebenyi, A. Nouhi, "A Programmable, Multi-Function Processor," S.P.I.E. Tech. Symp. East, Washington, D.C., April 1979.

4. S. Kowel, "The Deft Camera," Optical Spectra, July 1980.

5. S. Kowel, D. Cleverly, A. Mahapatra, T. Szebenyi, P. Kornreich, "A Two-Dimensional Acoustic Processor," Int. Optical Computing Conference, pp. 35-39, 1978.

6. S. Kowel, D. Cleverly, A. Mahapatra, T. Szebenyi, P. Kornreich, S. Wanuga, "Two-Dimensional Photoacoustic Image Processing With Longitudinal Waves," Proc. Ultrasonics Symp. pp. 258-262, 1978.

7. H. Gautier, P. Ledu and C. Maerfeld, "Two-Dimensional Acoustic Fourier Transform of Optical Images," Proc. Ultrasonics Symp. 1979.

8. G. Nagy, "State of the Art in Pattern Recognition," Proc. IEEE, Vol. 56, No. 5, pp. 836-862, 1968.

9. S. Dudani, K. Breeding and R. McGhee, "Aircraft Identification by Moment Invariants," IEEE Trans. Computers, Vol. C-26, No. 1, pp. 39-45, 1977.

10. J. McLaughlin and J. Raviv, "Nth Order Autocorrelations in Pattern Recognition," IEEE Int. Symp. on Information Theory, Aug. 1967.

11. R. Pickholtz, "Investigation of Linear Transformations for Automatic Cartographic Analysis," Interim reports No. 1, 2 and 3, Contract No. DAAK 70-78-C-0045, 1978, 1979.

12. M. Hu, "Visual Pattern Recognition by Moment Invariants," IRE Trans. on Information Theory, pp. 179-187, Feb. 1962.

13. M. Teague, "Image Analysis via the General Theory of Moments," J. Opt. Soc. Am., Vol. 70, No. 8, pp. 920-930, Aug. 1980.

14. P. Chen and W. Seemuller, "Signal Signatures of Topographic Features Using Analog Technology," Report No. ETL-0185, U.S. Army Engineer Topographic Laboratories, Ft. Belvoir, Va. 1979.

15. G. Schreiber, "Bandwidth Requirements for Walsh Functions," Proc. Walsh Function Symp., pp. 46-51, 1970.

16. A. VanderLugt, "Signal Detection by Complex Spatial Filtering," IEEE Trans. on Information Theory, Vol. IT-10, pp. 139-145, April 1964.

17. A. VanderLugt, "Coherent Optical Processing," Proc. IEEE, Vol. 62, No. 10, pp. 1300-1319, Oct 1974.

18. S. Craig and A. Moyer, "Image Alignment and Correlation System," Final Report No. ETL-0237 for U.S. Army Contract DAAK70-78-C-0217, July 1980.

19. D. Casasent and A. Furman, "Sources of Correlation Degradation," Applied Optics, Vol. 16, No. 6, pp. 1652-1661, June 1977.

20. G. Lendaris and G. Stanley, "Diffraction-Pattern Sampling for Automatic Pattern Recognition," Proc, IEEE, Vol. 58, No. 2, pp. 198-216, Feb. 1970.

21. B. Pernick, R. Kopp, J. Lisa, J. Mendelsohn, H. Stone and P. Wohlers, "Screening of Cervical Cytological Samples Using Coherent OpticalProcessing. Part 1," Applied Optics, Vol. 17, No. 1, pp. 21-34, Jan. 1978.

22. R. Gordon and G. Herman, "Three-Dimensional Reconstruction from Projections: A Review of Algorithms," Int. Rev. of Cytology, Vol, 38, pp. 111-151, 1974.

23. D. Casasent and D. Psaltis, "New Optical Transforms for Pattern Recognition," Proc. IEEE, Vol. 65, No. 1, pp. 77-84, Jan. 1977.

24. D. Casasent and D. Psaltis, "Position, Rotation and Scale Invariant Optical Correlation," Applied Optics, Vol, 15, No. 7, pp. 1795-1799, July 1976.

25. D. Casasent and M. Kraus, "Polar Camera for Space-Variant Pattern Recognition," Applied Optics, Vol, 17, No. 10, pp. 1559-1561, May 1978.

26. D. Psaltis and D. Casasent, "Deformation Invariant Optical Processors Using Coordinate Transformations," Applied Optics, Vol. 16, No. 8, pp. 2288-2292, Aug. 1977.

27. D. Casasent and D. Psaltis, "Multiple-Invariant Space - Variant Optical Processors," Applied Optics, Vol. 17, No. 4, pp. 655-659, Feb. 1978.

28. J. Volder, "The Cordic Trigonometric Computing Technique," IRE Trans. Comp., pp. 330-334, Sept. 1959.

29. M. Teague, "Optical Calculation of Irradiance Moments," Applied Optics, Vol. 19, No. 8, pp. 1353-1356, April 1980.

30. E. Isaacson and H. Keller, Analysis of Numerical Methods, John Wiley and Sons, 1966, Chapter 5, Section 5.1.

31. A. Mahapatra, S. Kowel, P. Kornreich, M. Mehter, "Spurious Acoustic Modes in Two-Dimensional Fourier Transform Devices," IEEE Trans. on Sonics and Ultrasonics, Vol, SU-25, No. 6, Nov. 1978.

APPENDIX A - Method of Invariant Fourier Signatures Assembly Code Listing

This Appendix consists of a listing of the assembly language program which computes invariant Fourier signatures. This program was written to run on the Deft Laboratories' microprocessor-based test bed. All addresses and opcodes are hexadecimal. In the operand column of the statements the following symbols are used:

$	Hexadecimal Prefix
%	Binary Prefix
H	Hexadecimal Postfix
D	Decimal Postfix
B	Binary Postfix
#	Denotes Immediate Addressing Mode

The entry address for this program is $2000.

```
1800                ORG   $1800
1800      TEMP1     RMB   1
1801      TEMP2     RMB   1
1802      TEMP3     RMB   1
1803      TEMP4     RMB   1
1804      TEMP5     RMB   1
1805      TEMP6     RMB   1
1806      BCD1      RMB   1
1807      BCD2      RMB   1
1824                ORG   $1824
1824      COR1      RMB   1
1825      COR2      RMB   1
1826      COR3      RMB   1
1827      COR4      RMB   3
182A      COR9      RMB   1
182B      COR10     RMB   5
1830      UI1       RMB   1
1831      UI2       RMB   1
1832      VJ1       RMB   1
1833      VJ2       RMB   2
1835      DTIME     RMB   1
183D                ORG   $183D
183D      NSAMP     RMB   1
183E      LOGS      RMB   18
1850      PUSHST    RMB   1
18B3                ORG   $18B3
18B3      FFTN      RMB   2
18E0                ORG   $18E0
18E0      REF       RMB   8
18E8      FIRST     RMB   1
18E9      CIR       RMB   1
18EA      M         RMB   1
18EB      N         RMB   1
18EC      I         RMB   1
18ED      P         RMB   1
18EE      TAG       RMB   1
18EF      THETA     RMB   2
18F1      DTHETA    RMB   2
18F3      PHI       RMB   2
18F5      DPHI      RMB   2
18F7      STACKS    RMB   2
18F9      RUN       RMB   2
18FB      RUN1      RMB   2
18FD      STOP      RMB   2
18FF      INT       RMB   4
1903      S1        RMB   8
190B      S2        RMB   8
1913      S3        RMB   8
191B      AVG       RMB   6
1921      POS       RMB   3
1924      NEG       RMB   3
2A3F      PASC      EQU   $2A3F
2A17      IDEX      EQU   $2A17
1385      ROT15     EQU   $1385
1376      VECT15    EQU   $1376
29AE      BI$BCD    EQU   $29AE
2A4C      RDDEFT    EQU   $2A4C
```

```
29FE                TUNE     EQU     $29FE
2999                DELAY1   EQU     $2999
292F                TRAPIN   EQU     $292F
2960                TRAPFX   EQU     $2960
1000                FFT      EQU     $1000
2A2A                WRITE    EQU     $2A2A
8000                MATH     EQU     $8000
290C                PUSH42   EQU     $290C
28EF                PUSH41   EQU     $28EF
28A3                PULL4    EQU     $28A3
2700                RCORR1   EQU     $2700
277F                RCORR2   EQU     $277F
0FFF                FFTVT    EQU     $FFF
0000                R0       EQU     $0
0000                         ORG     $0
0000   39                    FCB     $39
2000                         ORG     $2000
                    *
                    *        FEATURE EXTRACTION PROGRAM
                    *        DEFT/PROJECTION/FFT METHOD
2000   CE 18 E0     FEAT#1   LDX     #REF             CLEAR REF(I)
2003   C6 08                 LDAB    #8
2005   6F 00        FE1      CLR     0,X
2007   08                    INX
2008   5A                    DECB
2009   26 FA                 BNE     FE1
200B   7F 18 E8              CLR     FIRST
200E   BD FD A6              JSR     $FDA6
2011   CE 25 7D              LDX     #LINE1
2014   C6 13                 LDAB    #19D
2016   BD 2A 3F              JSR     PASC
2019   BD FD A6              JSR     $FDA6
201C   CE 25 90              LDX     #LINE2
201F   C6 13                 LDAB    #19D
2021   BD 2A 3F              JSR     PASC
2024   BD FD A6              JSR     $FDA6
2027   BD FD A6              JSR     $FDA6
202A   CE 25 A3              LDX     #LINE3
202D   C6 15                 LDAB    #21D
202F   BD 2A 3F              JSR     PASC
2032   BD FD 36              JSR     $FD36            INCH
2035   84 0F                 ANDA    #$0F
2037   B7 18 3E              STAA    LOGS
203A   C6 01                 LDAB    #1
203C   4A           FE2      DECA
203D   2D 03                 BLT     FE3
203F   58                    ASLB
2040   20 FA                 BRA     FE2
2042   F7 18 3D     FE3      STAB    NSAMP
2045   BD FD A6     FE4      JSR     $FDA6
2048   CE 25 B8              LDX     #LINE4
204B   C6 08                 LDAB    #8
204D   BD 2A 3F              JSR     PASC
2050   BD FD 36              JSR     $FD36            INCH
2053   5F                    CLRB
2054   81 59                 CMPA    #$59
2056   26 01                 BNE     FE5
```

```
2058  5C                   INCB                     YES
2059  F7 18 E9   FE5       STAB      CIR
205C  7D 18 E8             TST       FIRST
205F  27 02                BEQ       FE6
2061  20 46                BRA       FE11
2063  7D 18 E9   FE6       TST       CIR
2066  27 12                BEQ       FE7
2068  BD FD A6             JSR       $FDA6
206B  CE 25 C0             LDX       #LINE6
206E  C6 15                LDAB      #21D
2070  BD 2A 3F             JSR       PASC
2073  86 05                LDAA      #5
2075  B7 18 EA             STAA      M
2078  20 19                BRA       FE8
207A  BD FD A6   FE7       JSR       $FDA6
207D  CE 25 C0             LDX       #LINE6
2080  C6 14                LDAB      #20D
2082  BD 2A 3F             JSR       PASC
2085  BD FD 36             JSR       $FD36          INCH
2088  81 5A                CMPA      #$5A
208A  27 EE                BEQ       FE7
208C  84 0F                ANDA      #$F
208E  27 EA                BEQ       FE7
2090  B7 18 EA             STAA      M
2093  36         FE8       PSHA
2094  86 80                LDAA      #$80
2096  5F                   CLRB
2097  30                   TSX
2098  6D 00      FE9       TST       0,X
209A  27 06                BEQ       FE10
209C  6A 00                DEC       0,X
209E  47                   ASRA
209F  56                   RORB
20A0  20 F6                BRA       FE9
20A2  B7 18 F5   FE10      STAA      DPHI
20A5  F7 18 F6             STAB      DPHI+1
20A8  32                   PULA
20A9  BD FD A6   FE11      JSR       $FDA6
20AC  CE 25 D5             LDX       #LINE7
20AF  C6 13                LDAB      #19D
20B1  BD 2A 3F             JSR       PASC
20B4  BD FD 36             JSR       $FD36          INCH
20B7  81 5A                CMPA      #$5A
20B9  27 14                BEQ       FE12A
20BB  84 0F                ANDA      #$F
20BD  B7 18 EB             STAA      N
20C0  36                   PSHA
20C1  86 80                LDAA      #$80
20C3  5F                   CLRB
20C4  30                   TSX
20C5  6D 00      FE12      TST       0,X
20C7  27 0E                BEQ       FE13
20C9  6A 00                DEC       0,X
20CB  47                   ASRA
20CC  56                   RORB
20CD  20 F6                BRA       FE12
20CF  7D 18 E9   FE12A     TST       CIR
```

```
20D2  2E D5             BGT    FE11
20D4  36                PSHA
20D5  4F                CLRA
20D6  5F                CLRB
20D7  B7 18 F1   FE13   STAA   DTHETA
20DA  F7 18 F2          STAB   DTHETA+1
20DD  32                PULA
20DE  BD FD A6          JSR    $FDA6
20E1  CE 25 E8          LDX    $LINE8
20E4  C6 0E             LDAB   #14D
20E6  BD 2A 3F          JSR    PASC
20E9  BD FD 36          JSR    $FD36          INCH
20EC  84 0F             ANDA   #$F
20EE  B7 18 EE          STAA   TAG
20F1  7F 18 F3          CLR    PHI            PHI=0
20F4  7F 18 F4          CLR    PHI+1
20F7  CE C0 00          LDX    #$C000
20FA  FF 18 EF          STX    THETA          THETA=-PI/2
20FD  7F 18 EC          CLR    I
2100  CE 00 00          LDX    #0
2103  FF 19 1B          STX    AVG
2106  FF 19 1D          STX    AVG+2
2109  FF 19 1F          STX    AVG+4
210C  BF 18 F7          STS    STACKS
210F  8E 0F FF          LDS    #FFTVT
2112  CE 00 00   FE14   LDX    #0             LOOP: EACH CONTOUR
2115  FF 18 FF          STX    INT
2118  FF 19 01          STX    INT+2
211B  7F 18 ED          CLR    P
211E  7F 18 E8          CLR    FIRST
2121  F6 18 EC   FE15   LDAB   I              LOOP: EACH SAMPLE
2124  4F                CLRA
2125  58                ASLB
2126  49                ROLA
2127  CE 26 1A          LDX    #RVECT
212A  BD 2A 17          JSR    IDEX
212D  A6 00             LDAA   0,X            GET R(I) RADIUS
212F  B7 18 24          STAA   COR1
2132  A6 01             LDAA   1,X
2134  B7 18 25          STAA   COR2
2137  B6 18 EF          LDAA   THETA          ANGLE
213A  F6 18 F0          LDAB   THETA+1
213D  FB 18 F4          ADDB   PHI+1
2140  B9 18 F3          ADCA   PHI
2143  8B 40             ADDA   #$40           ADJUST QUADRANT
2145  2B 10             BMI    FE16
2147  B7 18 00          STAA   TEMP1
214A  F7 18 01          STAB   TEMP2
214D  4F                CLRA
214E  5F                CLRB
214F  F0 18 01          SUBB   TEMP2
2152  B2 18 00          SBCA   TEMP1
2155  20 09             BRA    FE17
2157  5D         FE16   TSTB
2158  26 06             BNE    FE17
215A  81 80             CMPA   #$80
215C  26 02             BNE    FE17
```

```
215E 4F               CLRA
215F 5F               CLRB
2160 8B C0     FE17   ADDA   #$C0
2162 B7 18 2A         STAA   COR9
2165 F7 18 2B         STAB   COR10
2168 7F 18 26         CLR    COR3
216B 7F 18 27         CLR    COR4
216E BD 13 85         JSR    ROT15          POLAR TO RECTANGULAR
2171 B6 18 24         LDAA   COR1
2174 F6 18 25         LDAB   COR2
2177 CE 00 05         LDX    #5
217A 47        FE18   ASRA
217B 56               RORB
217C 09               DEX
217D 26 FB            BNE    FE18
217F FB 26 17         ADDB   XZERO+1
2182 B9 26 16         ADCA   XZERO
2185 B7 18 02         STAA   TEMP3
2188 F7 18 03         STAB   TEMP4
218B BD 29 AE         JSR    BI$BCD
218E FE 18 06         LDX    BCD1
2191 FF 18 30         STX    UI1
2194 B6 18 26         LDAA   COR3
2197 F6 18 27         LDAB   COR4
219A CE 00 05         LDX    #5
219D 47        FE19   ASRA
219E 56               RORB
219F 09               DEX
21A0 26 FB            BNE    FE19
21A2 FB 26 19         ADDB   YZERO+1
21A5 B9 26 18         ADCA   YZERO
21A8 B7 18 02         STAA   TEMP3
21AB F7 18 03         STAB   TEMP4
21AE BD 29 AE         JSR    BI$BCD
21B1 FE 18 06         LDX    BCD1
21B4 FF 18 32         STX    VJ1
21B7 7D 18 E8         TST    FIRST
21BA 27 0E            BEQ    FE19A
21BC BD 2A 4C         JSR    RDDEFT
21BF BD 13 76         JSR    VECT15         MAG
21C2 BD 29 2F         JSR    TRAPIN         INTEGRATE
21C5 20 06            BRA    FE19B
21C7 7E 21 21  FE19C  JMP    FE15
21CA 7C 18 E8  FE19A  INC    FIRST
21CD BD 29 FE  FE19B  JSR    TUNE
21D0 86 00            LDAA   #0
21D2 B7 18 35         STAA   DTIME
21D5 BD 29 99         JSR    DELAY1
21D8 7D 18 E9         TST    CIR
21DB 27 44            BEQ    FE20
21DD B6 18 EF         LDAA   THETA          CIR=1
21E0 F6 18 F0         LDAB   THETA+1
21E3 FB 18 F2         ADDB   DTHETA+1
21E6 B9 18 F1         ADCA   DTHETA
21E9 B7 18 EF         STAA   THETA
21EC F7 18 F0         STAB   THETA+1        THETA=THETA+DTHETA
21EF 7D 18 F0         TST    THETA+1
```

```
21F2   26 D3              BNE    FE19C
21F4   B6 18 EF           LDAA   THETA
21F7   8B C0              ADDA   #$C0
21F9   26 CC              BNE    FE19C
21FB   BD 29 60           JSR    TRAPFX       THETA=+PI/2
21FE   B6 19 02           LDAA   INT+3
2201   36                 PSHA
2202   B6 19 01           LDAA   INT+2
2205   36                 PSHA
2206   B6 19 00           LDAA   INT+1
2209   36                 PSHA
220A   B6 18 FF           LDAA   INT
220D   36                 PSHA
220E   B6 18 EC           LDAA   I
2211   81 1F              CMPA   #31D
2213   27 6A              BEQ    FE23
2215   7C 18 EC           INC    I
2218   CE C0 00           LDX    #$C000
221B   FF 18 EF           STX    THETA
221E   7E 21 12           JMP    FE14
2221   B6 18 EC    FE20   LDAA   I
2224   81 1F              CMPA   #31D
2226   27 18              BEQ    FE21
2228   7C 18 EC           INC    I
222B   B6 18 EF           LDAA   THETA
222E   F6 18 F0           LDAB   THETA+1
2231   FB 18 F2           ADDB   DTHETA+1
2234   B9 18 F1           ADCA   DTHETA
2237   B7 18 EF           STAA   THETA
223A   F7 18 F0           STAB   THETA+1
223D   7E 21 21           JMP    FE15
2240   BD 29 60    FE21   JSR    TRAPFX       I=31
2243   B6 19 02           LDAA   INT+3
2246   36                 PSHA
2247   B6 19 01           LDAA   INT+2
224A   36                 PSHA

224B   B6 19 00           LDAA   INT+1
224E   36                 PSHA
224F   B6 18 FF           LDAA   INT
2252   36                 PSHA
2253   B6 18 F3           LDAA   PHI
2256   F6 18 F4           LDAB   PHI+1
2259   FB 18 F6           ADDB   DPHI+1
225C   B9 18 F5           ADCA   DPHI
225F   B7 18 F3           STAA   PHI          PHI=PHI+DPHI
2262   F7 18 F4           STAB   PHI+1
2265   7D 18 F4           TST    PHI+1
2268   26 09              BNE    FE22
226A   B6 18 F3           LDAA   PHI
226D   80 80              SUBA   #$80
226F   26 02              BNE    FE22
2271   20 10              BRA    FE24
2273   CE C0 00    FE22   LDX    #$C000
2276   FF 18 EF           STX    THETA
2279   7F 18 EC           CLR    I
227C   7E 21 12           JMP    FE14
```

```
227F   B6 05         FE23    LDAA    #5
2281   20 03                 BRA     FE25
2283   B6 18 EA      FE24    LDAA    M
2286   B7 18 50      FE25    STAA    PUSHST
2289   F6 19 1B      FE25A   LDAB    AVG              AVG=AVG/2**M
228C   57                    ASRB
228D   F7 19 1B              STAB    AVG
2290   F6 19 1C              LDAB    AVG+1
2293   56                    RORB
2294   F7 19 1C              STAB    AVG+1
2297   F6 19 1D              LDAB    AVG+2
229A   56                    RORB
229B   F7 19 1D              STAB    AVG+2
229E   F6 19 1E              LDAB    AVG+3
22A1   56                    RORB
22A2   F7 19 1E              STAB    AVG+3
22A5   F6 19 1F              LDAB    AVG+4
22A8   56                    RORB
22A9   F7 19 1F              STAB    AVG+4
22AC   F6 19 20              LDAB    AVG+5
22AF   56                    RORB
22B0   F7 19 20              STAB    AVG+5
22B3   4A                    DECA
22B4   26 D3                 BNE     FE25A
22B6   30                    TSX
22B7   7F 19 21              CLR     POS
22BA   7F 19 22              CLR     POS+1
22BD   7F 19 23              CLR     POS+2
22C0   86 FF                 LDAA    #$FF
22C2   B7 19 24              STAA    NEG
22C5   B7 19 25              STAA    NEG+1
22C8   B7 19 26              STAA    NEG+2
22CB   A6 03         FE25B   LDAA    3,X              REMOVE MEAN
22CD   B0 19 20              SUBA    AVG+5
22D0   A7 03                 STAA    3,X
22D2   A6 02                 LDAA    2,X
22D4   B2 19 1F              SBCA    AVG+4
22D7   A7 02                 STAA    2,X
22D9   A6 01                 LDAA    1,X
22DB   B2 19 1E              SBCA    AVG+3
22DE   A7 01                 STAA    1,X
22E0   A6 00                 LDAA    0,X
22E2   B2 19 1D              SBCA    AVG+2
22E5   A7 00                 STAA    0,X
22E7   2B 18                 BMI     FE25C            FIND SCALE FACTOR
22E9   BA 19 21              ORAA    POS
22EC   B7 19 21              STAA    POS
22EF   A6 01                 LDAA    1,X
22F1   BA 19 22              ORAA    POS+1
22F4   B7 19 22              STAA    POS+1
22F7   A6 02                 LDAA    2,X
22F9   BA 19 23              ORAA    POS+2
22FC   B7 19 23              STAA    POS+2
22FF   20 16                 BRA     FE25D
2301   B4 19 24      FE25C   ANDA    NEG
2304   B7 19 24              STAA    NEG
2307   A6 01                 LDAA    1,X
```

```
2309  B4 19 25          ANDA   NEG+1
230C  B7 19 25          STAA   NEG+1
230F  A6 02             LDAA   2,X
2311  B4 19 26          ANDA   NEG+2
2314  B7 19 26          STAA   NEG+2
2317  08       FE25D    INX
2318  08                INX
2319  08                INX
231A  08                INX
231B  8C 10 00          CPX    #FFTVT+1
231E  26 AB             BNE    FE25B
2320  B6 19 24          LDAA   NEG
2323  F6 19 25         .LDAB   NEG+1
2326  78 19 26          ASL    NEG+2
2329  CE 00 00          LDX    #0
232C  59       FE25E    ROLB
232D  49                ROLA
232E  24 04             BCC    FE25F
2330  08                INX
2331  0C                CLC
2332  20 F8             BRA    FE25E
2334  FF 19 24 FE25F    STX    NEG
2337  B6 19 21          LDAA   POS
233A  F6 19 22          LDAB   POS+1
233D  78 19 23          ASL    POS+2
2340  CE 00 00          LDX    #0
2343  59       FE25G    ROLB
2344  49                ROLA
2345  25 04             BCS    FE25H
2347  08                INX
2348  0D                SEC
2349  20 F8             BRA    FE25G
234B  FF 19 21 FE25H    STX    POS
234E  B6 19 22          LDAA   POS+1
2351  B0 19 25          SUBA   NEG+1
2354  2B 06             BMI    FE25I
2356  B6 19 25          LDAA   NEG+1
2359  B7 19 22          STAA   POS+1
235C  86 11    FE25I    LDAA   #17D
235E  B0 19 22          SUBA   POS+1
2361  2F 27             BLE    FE25L
2363  B7 19 22          STAA   POS+1
2366  30                TSX
2367  F6 19 22 FE25J    LDAB   POS+1          SCALE DOWN
236A  A6 00    FE25K    LDAA   0,X
236C  47                ASRA
236D  A7 00             STAA   0,X
236F  A6 01             LDAA   1,X
2371  46                RORA
2372  A7 01             STAA   1,X
2374  A6 02             LDAA   2,X
2376  46                RORA
2377  A7 02             STAA   2,X
2379  A6 03             LDAA   3,X
237B  46                RORA
237C  A7 03             STAA   3,X
237E  5A                DECB
```

```
237F  26 E9                BNE    FE25K
2381  08                   INX
2382  08                   INX
2383  08                   INX
2384  08                   INX
2385  8C 10 00             CPX    #FFTVT+1
2388  26 DD                BNE    FE25J
238A  B6 18 50   FE25L     LDAA   PUSHST
238D  5F                   CLRB
238E  BD 10 00             JSR    FFT              FFT
2391  B6 18 B3             LDAA   FFTN
2394  F6 18 B4             LDAB   FFTN+1
2397  47                   ASRA
2398  56                   RORB
2399  CE 00 00             LDX    #R0
239C  BD 2A 17             JSR    IDEX
239F  FF 18 FD             STX    STOP
23A2  CE 00 00             LDX    #R0
23A5  FF 18 F9             STX    RUN
23A8  CE 18 24   FE26      LDX    #COR1            PULL FFT/MAG/STORE
23AB  32                   PULA
23AC  A7 00                STAA   0,X
23AE  32                   PULA
23AF  A7 01                STAA   1,X
23B1  32                   PULA
23B2  A7 02                STAA   2,X
23B4  32                   PULA
23B5  A7 03                STAA   3,X
23B7  BD 13 76             JSR    VECT15
23BA  FE 18 F9             LDX    RUN
23BD  B6 18 24             LDAA   COR1
23C0  A7 00                STAA   0,X
23C2  B6 18 25             LDAA   COR2
23C5  A7 01                STAA   1,X
23C7  08                   INX
23C8  08                   INX
23C9  FF 18 F9             STX    RUN
23CC  BC 18 FD             CPX    STOP
23CF  26 D7                BNE    FE26
23D1  BE 18 F7             LDS    STACKS
23D4  BD FD A6             JSR    $FDA6
23D7  CE 25 F6             LDX    #LINE9
23DA  C6 0B                LDAB   #11D
23DC  BD 2A 3F             JSR    PASC
23DF  BD FD 36             JSR    $FD36            INCH
23E2  81 59                CMPA   #$59
23E4  26 50                BNE    FE29
23E6  CE 00 00             LDX    #R0
23E9  FF 18 F9             STX    RUN
23EC  86 01                LDAA   #1
23EE  B7 18 EC             STAA   I
23F1  BD FD A6             JSR    $FDA6
23F4  BD FD A6             JSR    $FDA6
23F7  FE 18 F9   FE27      LDX    RUN              PRINT SIGNATURE
23FA  A6 00                LDAA   0,X
23FC  B7 18 02             STAA   TEMP3
23FF  A6 01                LDAA   1,X
```

```
2401  B7 18 03          STAA   TEMP4
2404  08                INX
2405  08                INX
2406  FF 18 F9          STX    RUN
2409  BD 29 AE          JSR    BI$BCD
240C  CE 18 06          LDX    #BCD1
240F  BD 2A 2A          JSR    WRITE
2412  CE 18 07          LDX    #BCD2
2415  BD 2A 2A          JSR    WRITE
2418  86 20             LDAA   #$20
241A  BD FD 80          JSR    $FD80          BLANK
241D  BD FD 80          JSR    $FD80
2420  B6 18 EC          LDAA   I
2423  48                ASLA
2424  24 05             BCC    FE28
2426  BD FD A6          JSR    $FDA6
2429  86 01             LDAA   #1
242B  B7 18 EC   FE28   STAA   I
242E  FE 18 F9          LDX    RUN
2431  BC 18 FD          CPX    STOP
2434  26 C1             BNE    FE27
2436  BD FD A6   FE29   JSR    $FDA6
2439  BD FD A6          JSR    $FDA6
243C  CE 26 01          LDX    #LINE10
243F  C6 0B             LDAB   #11D
2441  BD 2A 3F          JSR    PASC
2444  BD FD A6          JSR    $FDA6
2447  86 01             LDAA   #1
2449  B7 18 EC          STAA   I
244C  B6 18 EC   FE30   LDAA   I
244F  B1 18 EE          CMPA   TAG
2452  26 53             BNE    FE33
2454  16                TAB
2455  8D 02             BSR    FE30A
2457  20 20             BRA    FE32
2459  B6 18 EA   FE30A  LDAA   M
245C  B7 18 00          STAA   TEMP1
245F  4F                CLRA
2460  58         FE31   ASLB
2461  49                ROLA
2462  7A 18 00          DEC    TEMP1
2465  26 F9             BNE    FE31
2467  58                ASLB
2468  49                ROLA
2469  CE 00 00          LDX    #R0
246C  BD 2A 17          JSR    IDEX           X=R0+I*2**(M+1)
246F  FF 18 FB          STX    RUN1
2472  CE 00 00          LDX    #R0
2475  FF 18 F9          STX    RUN
2478  39                RTS
2479  FE 18 F9   FE32   LDX    RUN            STORE NEW SIGNATURE IN R(I
247C  A6 00             LDAA   0,X
247E  E6 01             LDAB   1,X
2480  08                INX
2481  08                INX
2482  FF 18 F9          STX    RUN
2485  FE 18 FB          LDX    RUN1
```

```
2488  A7 00             STAA   0,X
248A  E7 01             STAB   1,X
248C  08                INX
248D  08                INX
248E  FF 18 FB          STX    RUN1
2491  FE 18 F9          LDX    RUN
2494  BC 18 FD          CPX    STOP
2497  26 E0             BNE    FE32
2499  CE 18 DF          LDX    #REF-1
249C  F6 18 EC          LDAB   I
249F  4F                CLRA
24A0  BD 2A 17          JSR    IDEX
24A3  86 01             LDAA   #1
24A5  A7 00             STAA   0,X         SET REF(I)=1
24A7  F6 18 EC   FE33   LDAB   I
24AA  4F                CLRA
24AB  CE 18 DF          LDX    #REF-1
24AE  BD 2A 17          JSR    IDEX
24B1  A6 00             LDAA   0,X
24B3  81 01             CMPA   #1
24B5  27 09             BEQ    FE34
24B7  7F 18 06          CLR    BCD1
24BA  7F 18 07          CLR    BCD2
24BD  7E 25 14          JMP    FE37
24C0  F6 18 EC   FE34   LDAB   I           REF(I)=1
24C3  8D 94             BSR    FE30A
24C5  4F                CLRA
24C6  C6 18             LDAB   #24D
24C8  CE 19 03          LDX    #S1
24CB  A7 00      FE35   STAA   0,X         S1,S2,S3 NFPZ
24CD  08                INX
24CE  5A                DECB
24CF  26 FA             BNE    FE35
24D1  86 80             LDAA   #$80
24D3  B7 19 0A          STAA   S1+7
24D6  B7 19 12          STAA   S2+7
24D9  B7 19 1A          STAA   S3+7
24DC  FE 18 F9   FE36   LDX    RUN         CORRELATE R(0) WITH R(I)
24DF  A6 00             LDAA   0,X
24E1  B7 18 02          STAA   TEMP3
24E4  A6 01             LDAA   1,X
24E6  B7 18 03          STAA   TEMP4
24E9  08                INX
24EA  08                INX
24EB  FF 18 F9          STX    RUN
24EE  FE 18 FB          LDX    RUN1
24F1  A6 00             LDAA   0,X
24F3  B7 18 04          STAA   TEMP5
24F6  A6 01             LDAA   1,X
24F8  B7 18 05          STAA   TEMP6
24FB  08                INX
24FC  08                INX
24FD  FF 18 FB          STX    RUN1
2500  CE 18 02          LDX    #TEMP3
2503  BD 27 00          JSR    RCORR1      ACCUMULATE
2506  FE 18 F9          LDX    RUN
2509  BC 18 FD          CPX    STOP
```

```
250C  26 CE              BNE    FE36
250E  BD 27 7F           JSR    RCORR2            S1/(S2*S3)**0.5
2511  BD 29 AE           JSR    BI$BCD
2514  B6 18 EC  FE37     LDAA   I
2517  8A 30              ORAA   #$30
2519  BD FD 80           JSR    $FD80             PRINT I
251C  86 29              LDAA   #$29
251E  BD FD 80           JSR    $FD80             PRINT ')'
2521  86 20              LDAA   #$20
2523  BD FD 80           JSR    $FD80             BLANK
2526  86 30              LDAA   #$30
2528  BD FD 80           JSR    $FD80             PRINT 0
252B  86 2E              LDAA   #$2E
252D  BD FD 80           JSR    $FD80             PRINT '.'
2530  CE 18 06           LDX    #BCD1
2533  BD 2A 2A           JSR    WRITE             CORR
2536  CE 18 07           LDX    #BCD2
2539  BD 2A 2A           JSR    WRITE
253C  86 20              LDAA   #$20
253E  BD FD 80           JSR    $FD80             BLANK
2541  BD FD 80           JSR    $FD80
2544  B6 18 EC           LDAA   I
2547  81 04              CMPA   #4
2549  26 03              BNE    FE38
254B  BD FD A6           JSR    $FDA6
254E  7C 18 EC  FE38     INC    I
2551  B6 18 EC           LDAA   I
2554  81 09              CMPA   #9
2556  26 1F              BNE    FE39
2558  BD FD A6           JSR    $FDA6
255B  CE 26 0C           LDX    #LINE11
255E  C6 0A              LDAB   #10D
2560  BD 2A 3F           JSR    PASC
2563  BD FD 36           JSR    $FD36             INCH
2566  81 59              CMPA   #$59
2568  26 10              BNE    FE40
256A  B6 18 EA           LDAA   M
256D  81 05              CMPA   #5
256F  27 03              BEQ    FE38A
2571  7E 20 A9           JMP    FE11
2574  7E 20 45  FE38A    JMP    FE4
2577  7E 24 4C  FE39     JMP    FE30
257A  7E FE 2D  FE40     JMP    $FE2D             FANTOM
257D            LINE1    EQU    *
257D  46 45 41           FCC    'FEATURE '
2585  52 45 43           FCC    'RECOGNITION'
2590            LINE2    EQU    *
2590  44 45 46           FCC    'DEFT/PROJECTION/'
25A0  46 46 54           FCC    'FFT'
25A3            LINE3    EQU    *
25A3  53 41 4D           FCC    'SAMPLE AVERAGE'
25B1  20 49 53           FCC    ' IS 2**'
25B8            LINE4    EQU    *
25B8  43 49 52           FCC    'CIRCLES?'
25C0            LINE6    EQU    *
25C0  52 45 46           FCC    'REFERENCE LENGTH'
25D0  3D 32 2A           FCC    '=2**5'
```

```
25D5                LINE7   EQU    *
25D5  44 45 4C              FCC    'DELTA THETA=-PI/'
25E5  32 2A 2A              FCC    '2**'
25E8                LINE8   EQU    *
25E8  52 45 46              FCC    'REFERENCE TAG='
25F6                LINE9   EQU    *
25F6  53 49 47              FCC    'SIGNATURE? '
2601                LINE10  EQU    *
2601  43 4F 52              FCC    'CORRELATION'
260C                LINE11  EQU    *
260C  43 4F 4E              FCC    'CONTINUE? '
2616                XZERO   EQU    *
2616  0F 11                 FCB    $F,$11             3857
2618                YZERO   EQU    *
2618  0D AA                 FCB    $D,$AA             3498
261A                RVECT   EQU    *
261A  07 2B 07              FCB    $7,$2B,$7,$C8,$8,$72,$9,$2B
2622  09 F4 0A              FCB    $9,$F4,$A,$CE,$B,$BA,$C,$BB
262A  0D D1 0E              FCB    $D,$D1,$E,$FF,$10,$47,$11,$AB
2632  13 2E 14              FCB    $13,$2E,$14,$D1,$16,$99,$18,$87
263A  1A A0 1C              FCB    $1A,$A0,$1C,$E6,$1F,$5F,$22,$D
2642  24 F6 28              FCB    $24,$F6,$28,$1E,$2B,$8C,$2F,$45
264A  33 4F 37              FCB    $33,$4F,$37,$B1,$3C,$74,$41,$9E
2652  47 3A 4D              FCB    $47,$3A,$4D,$50,$53,$EB,$5B,$17
265A                        END
```

STATEMENTS =710

FREE BYTES =16551

NO ERRORS DETECTED

```
18B0                ORG     $18B0
18B0        BLKSF   RMB     1
18B1        M       RMB     1
18B2        TFLAG   RMB     1
18B3        N       RMB     2
18B5        LE      RMB     2
18B7        ANGLE   RMB     2
18B9        L       RMB     1
18BA        LE1     RMB     2
18BC        THETA   RMB     2
18BE        J       RMB     2
18C0        I       RMB     2
18C2        BUT1R   RMB     2
18C4        BUT1I   RMB     2
18C6        BUT2R   RMB     2
18C8        BUT2I   RMB     2
18CA        IP      RMB     2
18CC        II      RMB     2
18CE        NV2     RMB     2
18D0        NM1     RMB     2
18D2        K       RMB     2
18D4        JJ      RMB     2
18D6        BSN     RMB     2
18D8        POS     RMB     2
18DA        NEG     RMB     2
18DC        BSN1    RMB     2
1824                ORG     $1824
1824        COR1    RMB     1
1825        COR2    RMB     1
1826        COR3    RMB     1
1827        COR4    RMB     1
1828        DNUP1   RMB     1
1829        DNUP2   RMB     1
182A        COR9    RMB     1
182B        COR10   RMB     1
1800                ORG     $1800
1800        COR5    RMB     1
1801        COR6    RMB     1
1802        COR7    RMB     1
1803        COR8    RMB     1
1804        COR11   RMB     1
1805        COR12   RMB     1
1806        COR13   RMB     1
1807        COR14   RMB     1
1808        COR15   RMB     1
1809        COR16   RMB     1
180A        COR17   RMB     1
180B        COR18   RMB     1
180C        CFLAG   RMB     1
180D        COR19   RMB     2
180F        ITER    RMB     1
1000                ORG     $1000
            *       SUBROUTINE FFT
            *       REAL, IMAG ALTERNATE IN STACK
            *       ENTER WITH A ACCUM=M FOR
            *       2**M FFT; B=0 FOR FORWARD FFT,
            *       B=1 FOR INVERSE FFT
```

```
1000  7F 18 B0  FFT   CLR   BLKSF
1003  CE 18 B1        LDX   #M
1006  A7 00           STAA  0,X         M
1008  E7 01           STAB  1,X         TFLAG
100A  C6 01           LDAB  #1
100C  4C              INCA
100D  4C              INCA
100E  0C              CLC
100F  6F 02           CLR   2,X
1011  59        F1    ROLB
1012  24 04           BCC   F2
1014  6F 03           CLR   3,X
1016  09              DEX
1017  5C              INCB
1018  4A        F2    DECA
1019  26 F6           BNE   F1
101B  E7 03           STAB  3,X         N=2**(M+2)
101D  CE 18 B1        LDX   #M
1020  A6 02           LDAA  2,X
1022  E6 03           LDAB  3,X
1024  58              ASLB
1025  49              ROLA
1026  A7 04           STAA  4,X         LE=2**(M+3)
1028  E7 05           STAB  5,X
102A  A6 00           LDAA  0,X
102C  C6 80           LDAB  #$80
102E  6F 07           CLR   7,X
1030  0C              CLC
1031  56        F3    RORB
1032  24 04           BCC   F4
1034  6F 06           CLR   6,X
1036  08              INX
1037  56              RORB
1038  4A        F4    DECA
1039  26 F6           BNE   F3
103B  E7 06           STAB  6,X         ANGLE=2**(15-M)
103D  CE 18 B1        LDX   #M
1040  6D 01           TST   1,X
1042  2E 0A           BGT   F4A
1044  4F              CLRA              FORWARD FFT
1045  5F              CLRB
1046  E0 07           SUBB  7,X
1048  A2 06           SBCA  6,X
104A  A7 06           STAA  6,X         ANGLE=-ANGLE
104C  E7 07           STAB  7,X
104E  86 01     F4A   LDAA  #1
1050  A7 08           STAA  8,X
1052  FE 18 B3  F5    LDX   N           MAIN LOOP
1055  BD 12 AC        JSR   BSCALE
1058  BB 18 B0        ADDA  BLKSF
105B  B7 18 B0        STAA  BLKSF       BLOCK FLOAT POINT SCALE F
105E  CE 18 B5        LDX   #LE
1061  A6 00           LDAA  0,X
1063  E6 01           LDAB  1,X
1065  44              LSRA
1066  56              RORB
1067  A7 00           STAA  0,X         LE=LE/2
```

```
1069  E7 01              STAB   1,X
106B  44                 LSRA
106C  56                 RORB
106D  A7 05              STAA   5,X
106F  E7 06              STAB   6,X          LE1=LE/2
1071  A6 02              LDAA   2,X
1073  E6 03              LDAB   3,X
1075  58                 ASLB
1076  49                 ROLA
1077  A7 02              STAA   2,X          ANGLE=2*ANGLE
1079  E7 03              STAB   3,X
107B  6F 07              CLR    7,X          THETA=0
107D  6F 08              CLR    8,X
107F  6F 09              CLR    9,X
1081  86 04              LDAA   #4
1083  A7 0A              STAA   10D,X        J=4
1085  30          F6     TSX
1086  09                 DEX
1087  09                 DEX
1088  FF 18 C2           STX    BUT1R
108B  CE 18 BE           LDX    #J
108E  A6 00              LDAA   0,X
1090  E6 01              LDAB   1,X
1092  A7 0E              STAA   14D,X        II=J
1094  E7 0F              STAB   15D,X
1096  EB 05              ADDB   5,X
1098  A9 04              ADCA   4,X
109A  A7 02              STAA   2,X          I=J+STKPTR-1
109C  E7 03              STAB   3,X
109E  CE 18 BA    F7     LDX    #LE1
10A1  A6 00              LDAA   0,X
10A3  E6 01              LDAB   1,X
10A5  EB 07              ADDB   7,X
10A7  A9 06              ADCA   6,X
10A9  A7 10              STAA   16D,X        IP=I+LE1
10AB  E7 11              STAB   17D,X
10AD  FE 18 C0           LDX    I            BUTTERFLY
10B0  A6 00              LDAA   0,X
10B2  B7 18 C2           STAA   BUT1R
10B5  A6 01              LDAA   1,X
10B7  B7 18 C3           STAA   BUT1R+1
10BA  A6 02              LDAA   2,X
10BC  B7 18 C4           STAA   BUT1I
10BF  A6 03              LDAA   3,X
10C1  B7 18 C5           STAA   BUT1I+1
10C4  FE 18 CA           LDX    IP
10C7  A6 00              LDAA   0,X
10C9  B7 18 C6           STAA   BUT2R
10CC  A6 01              LDAA   1,X
10CE  B7 18 C7           STAA   BUT2R+1
10D1  A6 02              LDAA   2,X
10D3  B7 18 C8           STAA   BUT2I
10D6  A6 03              LDAA   3,X
10D8  B7 18 C9           STAA   BUT2I+1
10DB  CE 18 C2           LDX    #BUT1R
10DE  A6 00              LDAA   0,X
10E0  E6 01              LDAB   1,X
```

```
10E2  EB 05        ADDB   5,X
10E4  A9 04        ADCA   4,X
10E6  B7 18 24     STAA   COR1
10E9  F7 18 25     STAB   COR2
10EC  A6 02        LDAA   2,X
10EE  E6 03        LDAB   3,X
10F0  EB 07        ADDB   7,X
10F2  A9 06        ADCA   6,X
10F4  B7 18 26     STAA   COR3
10F7  F7 18 27     STAB   COR4
10FA  7F 18 2A     CLR    COR9
10FD  7F 18 2B     CLR    COR10
1100  BD 13 85     JSR    ROT15
1103  FE 18 C0     LDX    I
1106  B6 18 24     LDAA   COR1
1109  A7 00        STAA   0,X
110B  B6 18 25     LDAA   COR2
110E  A7 01        STAA   1,X
1110  B6 18 26     LDAA   COR3
1113  A7 02        STAA   2,X
1115  B6 18 27     LDAA   COR4
1118  A7 03        STAA   3,X
111A  CE 18 C2     LDX    #BUT1R
111D  A6 00        LDAA   0,X
111F  E6 01        LDAB   1,X
1121  E0 05        SUBB   5,X
1123  A2 04        SBCA   4,X
1125  B7 18 24     STAA   COR1
1128  F7 18 25     STAB   COR2
112B  A6 02        LDAA   2,X
112D  E6 03        LDAB   3,X
112F  E0 07        SUBB   7,X
1131  A2 06        SBCA   6,X
1133  B7 18 26     STAA   COR3
1136  F7 18 27     STAB   COR4
1139  FE 18 BC     LDX    THETA
113C  FF 18 2A     STX    COR9
113F  BD 13 85     JSR    ROT15
1142  FE 18 CA     LDX    IP
1145  B6 18 24     LDAA   COR1
1148  A7 00        STAA   0,X
114A  B6 18 25     LDAA   COR2
114D  A7 01        STAA   1,X
114F  B6 18 26     LDAA   COR3
1152  A7 02        STAA   2,X
1154  B6 18 27     LDAA   COR4
1157  A7 03        STAA   3,X
1159  CE 18 B3     LDX    #N
115C  A6 02        LDAA   2,X
115E  E6 03        LDAB   3,X
1160  EB 0E        ADDB   14D,X
1162  A9 0D        ADCA   13D,X
1164  A7 0D        STAA   13D,X
1166  E7 0E        STAB   14D,X          I=I+LE
1168  A6 02        LDAA   2,X
116A  E6 03        LDAB   3,X
116C  EB 1A        ADDB   26D,X
```

```
116E  A9 19              ADCA   25D,X
1170  A7 19              STAA   25D,X      II=II+LE
1172  E7 1A              STAB   26D,X
1174  E0 01              SUBB   1,X        II-N
1176  A2 00              SBCA   0,X
1178  2D 07              BLT    F7A
117A  2E 08              BGT    F8
117C  5D                 TSTB
117D  27 02              BEQ    F7A
117F  20 03              BRA    F8
1181  7E 10 9E  F7A      JMP    F7
1184  A6 09     F8       LDAA   9,X
1186  E6 0A              LDAB   10D,X
1188  EB 05              ADDB   5,X
118A  A9 04              ADCA   4,X
118C  A7 09              STAA   9,X        THETA=THETA+ANGLE
118E  E7 0A              STAB   10D,X
1190  A6 0B              LDAA   11D,X
1192  E6 0C              LDAB   12D,X
1194  CB 04              ADDB   #4
1196  89 00              ADCA   #0
1198  A7 0B              STAA   11D,X      J=J+4
119A  E7 0C              STAB   12D,X
119C  E0 08              SUBB   8,X        J-LE1
119E  A2 07              SBCA   7,X
11A0  2D 07              BLT    F8A
11A2  2E 08              BGT    F9
11A4  5D                 TSTB
11A5  27 02              BEQ    F8A
11A7  20 03              BRA    F9
11A9  7E 10 85  F8A      JMP    F6
11AC  6C 06     F9       INC    6,X        L=L+1
11AE  A6 06              LDAA   6,X
11B0  B0 18 B1           SUBA   M          L-M
11B3  2E 03              BGT    F9A
11B5  7E 10 52           JMP    F5
11B8  A6 00     F9A      LDAA   0,X        BIT REVERSAL
11BA  E6 01              LDAB   1,X
11BC  47                 ASRA
11BD  56                 RORB
11BE  A7 1B              STAA   27D,X      NV2=N/2
11C0  E7 1C              STAB   28D,X
11C2  A6 00              LDAA   0,X
11C4  E6 01              LDAB   1,X
11C6  C0 04              SUBB   #4
11C8  82 00              SBCA   #0
11CA  A7 1D              STAA   29D,X      NM1=N-4
11CC  E7 1E              STAB   30D,X
11CE  6F 21              CLR    33D,X
11D0  86 04              LDAA   #4
11D2  A7 22              STAA   34D,X      JJ=4
11D4  6F 19              CLR    25D,X
11D6  A7 1A              STAA   26D,X      II=4
11D8  30                 TSX
11D9  08                 INX
11DA  08                 INX
11DB  FF 18 C0           STX    I          I=STKPTR+3
```

```
11DE   FF 18 BE          STX    J          J=I
11E1   CE 18 BA   F10    LDX    #LE1
11E4   A6 06             LDAA   6,X
11E6   E6 07             LDAB   7,X
11E8   E0 05             SUBB   5,X        I-J
11EA   A2 04             SBCA   4,X
11EC   2C 4A             BGE    F11
11EE   FE 18 BE          LDX    J
11F1   A6 00             LDAA   0,X
11F3   B7 18 C2          STAA   BUT1R
11F6   A6 01             LDAA   1,X
11F8   B7 18 C3          STAA   BUT1R+1
11FB   A6 02             LDAA   2,X
11FD   B7 18 C4          STAA   BUT1I
1200   A6 03             LDAA   3,X
1202   B7 18 C5          STAA   BUT1I+1
1205   FE 18 C0          LDX    I
1208   A6 00             LDAA   0,X
120A   E6 01             LDAB   1,X
120C   FE 18 BE          LDX    J
120F   A7 00             STAA   0,X
1211   E7 01             STAB   1,X
1213   FE 18 C0          LDX    I
1216   A6 02             LDAA   2,X
1218   E6 03             LDAB   3,X
121A   FE 18 BE          LDX    J
121D   A7 02             STAA   2,X
121F   E7 03             STAB   3,X
1221   FE 18 C0          LDX    I
1224   B6 18 C2          LDAA   BUT1R
1227   A7 00             STAA   0,X
1229   B6 18 C3          LDAA   BUT1R+1
122C   A7 01             STAA   1,X
122E   B6 18 C4          LDAA   BUT1I
1231   A7 02             STAA   2,X
1233   B6 18 C5          LDAA   BUT1I+1
1236   A7 03             STAA   3,X
1238   CE 18 BA   F11    LDX    #LE1
123B   A6 14             LDAA   20D,X
123D   E6 15             LDAB   21D,X
123F   A7 18             STAA   24D,X      K=NV2
1241   E7 19             STAB   25D,X
1243   E0 1B      F12    SUBB   27D,X      K-JJ
1245   A2 1A             SBCA   26D,X
1247   2C 24             BGE    F13
1249   A6 04             LDAA   4,X
124B   E6 05             LDAB   5,X
124D   E0 19             SUBB   25D,X
124F   A2 18             SBCA   24D,X
1251   A7 04             STAA   4,X        J=J-K
1253   E7 05             STAB   5,X
1255   A6 1A             LDAA   26D,X
1257   E6 1B             LDAB   27D,X
1259   E0 19             SUBB   25D,X
125B   A2 18             SBCA   24D,X
125D   A7 1A             STAA   26D,X      JJ=JJ-K
125F   E7 1B             STAB   27D,X
```

```
1261  A6 18           LDAA   24D,X
1263  E6 19           LDAB   25D,X
1265  47              ASRA
1266  56              RORB
1267  A7 18           STAA   24D,X        K=K/2
1269  E7 19           STAB   25D,X
126B  20 D6           BRA    F12
126D  A6 04     F13   LDAA   4,X
126F  E6 05           LDAB   5,X
1271  EB 19           ADDB   25D,X
1273  A9 18           ADCA   24D,X
1275  A7 04           STAA   4,X          J=J+K
1277  E7 05           STAB   5,X
1279  A6 06           LDAA   6,X
127B  E6 07           LDAB   7,X
127D  CB 04           ADDB   #4
127F  89 00           ADCA   #0
1281  A7 06           STAA   6,X          I=I+4
1283  E7 07           STAB   7,X
1285  A6 1A           LDAA   26D,X
1287  E6 1B           LDAB   27D,X
1289  EB 19           ADDB   25D,X
128B  A9 18           ADCA   24D,X
128D  A7 1A           STAA   26D,X        JJ=JJ+K
128F  E7 1B           STAB   27D,X
1291  A6 12           LDAA   18D,X
1293  E6 13           LDAB   19D,X
1295  CB 04           ADDB   #4
1297  89 00           ADCA   #0
1299  A7 12           STAA   18D,X        II=II+4
129B  E7 13           STAB   19D,X
129D  E0 17           SUBB   23D,X        II-NM1
129F  A2 16           SBCA   22D,X
12A1  2D 06           BLT    F15
12A3  2E 03           BGT    F14
12A5  5D              TSTB
12A6  27 01           BEQ    F15
12A8  39        F14   RTS
12A9  7E 11 E1  F15   JMP    F10
```

```
                        *       SUBROUTINE BSCALE
                        *       BFP SCALING OF DATA IN STACK
                        *       VECTOR CONSISTS OF N/2, 2 BYTE
                        *       NUMBERS. ENTER WITH N IN X REG.
                        *       RETURNS BLOCK SCALE FACTOR IN A REG.
12AC  FF 18 D6  BSCALE  STX     BSN
12AF  FF 18 DC          STX     BSN1
12B2  CE 00 00          LDX     #0
12B5  FF 18 D8          STX     POS
12B8  CE FF FF          LDX     #$FFFF
12BB  FF 18 DA          STX     NEG
12BE  30        BS1     TSX
12BF  08                INX
12C0  08                INX
12C1  08        BS1A    INX
12C2  08                INX
12C3  E6 01             LDAB    1,X
12C5  A6 00             LDAA    0,X
12C7  2B 0E             BMI     BS2
12C9  BA 18 D8          ORAA    POS
12CC  FA 18 D9          ORAB    POS+1
12CF  B7 18 D8          STAA    POS
12D2  F7 18 D9          STAB    POS+1
12D5  20 0C             BRA     BS0
12D7  B4 18 DA  BS2     ANDA    NEG
12DA  F4 18 DB          ANDB    NEG+1
12DD  B7 18 DA          STAA    NEG
12E0  F7 18 DB          STAB    NEG+1
12E3  B6 18 D6  BS0     LDAA    BSN
12E6  F6 18 D7          LDAB    BSN+1
12E9  C0 02             SUBB    #2
12EB  82 00             SBCA    #0
12ED  B7 18 D6          STAA    BSN
12F0  F7 18 D7          STAB    BSN+1
12F3  2E CC             BGT     BS1A
12F5  5D                TSTB
12F6  26 C9             BNE     BS1A
12F8  B6 18 DA          LDAA    NEG
12FB  F6 18 DB          LDAB    NEG+1
12FE  CE 00 00          LDX     #0
1301  58        BS4     ASLB
1302  49                ROLA
1303  24 03             BCC     BS5
1305  08                INX
1306  20 F9             BRA     BS4
1308  09        BS5     DEX
1309  09                DEX
130A  FF 18 DA          STX     NEG
130D  B6 18 D8          LDAA    POS
1310  F6 18 D9          LDAB    POS+1
1313  CE 00 00          LDX     #0
1316  0D                SEC
1317  59                ROLB
1318  20 01             BRA     BS6A
131A  58        BS6     ASLB
131B  49        BS6A    ROLA
131C  25 03             BCS     BS7
```

```
131E  08              INX
131F  20 F9           BRA    BS6
1321  09        BS7   DEX
1322  09              DEX
1323  FF 18 D8        STX    POS
1326  CE 18 D8        LDX    #POS
1329  E6 01           LDAB   1,X
132B  E0 03           SUBB   3,X
132D  2B 04           BMI    BS8
132F  A6 03           LDAA   3,X
1331  A7 01           STAA   1,X
1333  6D 01     BS8   TST    1,X           POS+1=SCALE FACTOR
1335  26 02           BNE    BS9
1337  4F              CLRA
1338  39              RTS                  BLOCK S.F.=0
1339  30        BS9   TSX
133A  08              INX
133B  08              INX
133C  08        BS9A  INX                  SCALE DATA
133D  08              INX
133E  B6 18 D9        LDAA   POS+1
1341  2B 10           BMI    BS11
1343  B7 18 D8        STAA   POS
1346  A6 00           LDAA   0,X
1348  E6 01           LDAB   1,X
134A  58        BS10  ASLB
134B  49              ROLA
134C  7A 18 D8        DEC    POS
134F  2E F9           BGT    BS10
1351  20 06           BRA    BS12
1353  A6 00     BS11  LDAA   0,X
1355  E6 01           LDAB   1,X
1357  47              ASRA
1358  56              RORB
1359  A7 00     BS12  STAA   0,X
135B  E7 01           STAB   1,X
135D  B6 18 DC        LDAA   BSN1
1360  F6 18 DD        LDAB   BSN1+1
1363  C0 02           SUBB   #2
1365  82 00           SBCA   #0
1367  B7 18 DC        STAA   BSN1
136A  F7 18 DD        STAB   BSN1+1
136D  2E CD           BGT    BS9A
136F  5D              TSTB
1370  26 CA           BNE    BS9A
1372  B6 18 D9        LDAA   POS+1
1375  39              RTS
```

```
                        *
                        *
                        *      CORDIC ALGORITHM ITER ITERATIONS
1376   86 0F     VECT15  LDAA   #15D          VECTOR: 15 ITERATIONS
1378   B7 18 0F          STAA   ITER
137B   7F 18 2A  VECT    CLR    COR9          VECTOR: ITER ITERATIONS
137E   7F 18 2B          CLR    COR10
1381   86 01             LDAA   #1
1383   20 07             BRA    LCOR0
1385   86 0F     ROT15   LDAA   #15D          ROTATION: 15 ITERATIONS
1387   B7 18 0F          STAA   ITER
138A   86 80     ROT     LDAA   #$80          ROTATION: ITER ITERATIONS
138C   CE 14 D9  LCOR0   LDX    #DLO
138F   FF 18 0D          STX    COR19
1392   CE 18 00          LDX    #COR5
1395   A7 0C             STAA   $C,X
1397   A6 2A             LDAA   $2A,X
1399   A7 0A             STAA   $A,X
139B   6F 0B             CLR    $B,X
139D   A6 24             LDAA   $24,X
139F   E6 25             LDAB   $25,X
13A1   BD 14 91          JSR    SCALE
13A4   A7 00             STAA   0,X
13A6   A6 26             LDAA   $26,X
13A8   E6 27             LDAB   $27,X
13AA   BD 14 91          JSR    SCALE
13AD   A7 01             STAA   1,X
13AF   A6 00             LDAA   0,X
13B1   A4 01             ANDA   1,X
13B3   A7 01             STAA   1,X
13B5   27 0C             BEQ    LCOR1
13B7   A6 25             LDAA   $25,X
13B9   A7 24             STAA   $24,X
13BB   6F 25             CLR    $25,X
13BD   A6 27             LDAA   $27,X
13BF   A7 26             STAA   $26,X
13C1   6F 27             CLR    $27,X
13C3   6F 02     LCOR1   CLR    2,X
13C5   6F 03             CLR    3,X
13C7   6F 08             CLR    8,X
13C9   6F 09             CLR    9,X
13CB   6F 28             CLR    $28,X
13CD   A6 24             LDAA   $24,X
13CF   84 E0             ANDA   #$E0
13D1   2C 01             BGE    LCOR2
13D3   43                COMA
13D4   E6 26     LCOR2   LDAB   $26,X
13D6   C4 E0             ANDB   #$E0
13D8   2C 01             BGE    LCOR3
13DA   53                COMB
13DB   E7 29     LCOR3   STAB   $29,X
13DD   C6 01             LDAB   #1
13DF   E7 00             STAB   0,X
13E1   AA 29             ORAA   $29,X
13E3   27 05             BEQ    LCOR3A
13E5   BD 14 BC          JSR    DOWN2
13E8   6A 00             DEC    0,X
```

```
13EA  6F 04     LCOR3A  CLR    4,X
13EC  A6 0F             LDAA   $F,X
13EE  A7 05             STAA   5,X
13F0  CE 15 32          LDX    #CTABLE
13F3  FF 18 06          STX    COR13
13F6  FE 18 06  LCOR4   LDX    COR13          MAIN LOOP
13F9  A6 00             LDAA   0,X
13FB  E6 01             LDAB   1,X
13FD  08                INX
13FE  08                INX
13FF  FF 18 06          STX    COR13
1402  FE 18 0D          LDX    COR19
1405  08                INX
1406  08                INX
1407  FF 18 0D          STX    COR19
140A  CE 18 00          LDX    #COR5
140D  BD 14 AB          JSR    TSTCFL
1410  2C 03             BGE    LCOR5
1412  BD 14 A4          JSR    COMP
1415  EB 2B     LCOR5   ADDB   $2B,X
1417  A9 2A             ADCA   $2A,X
1419  A7 2A             STAA   $2A,X
141B  E7 2B             STAB   $2B,X          THETA DONE
141D  A6 26             LDAA   $26,X
141F  E6 27             LDAB   $27,X
1421  BD 14 AB          JSR    TSTCFL
1424  2C 03             BGE    LCOR6
1426  BD 14 A4          JSR    COMP
1429  BD 14 D5  LCOR6   JSR    DOWN
142C  CE 18 00          LDX    #COR5
142F  EB 03             ADDB   3,X
1431  A9 02             ADCA   2,X
1433  A7 02             STAA   2,X
1435  E7 03             STAB   3,X            X DONE
1437  A6 24             LDAA   $24,X
1439  E6 25             LDAB   $25,X
143B  BD 14 AB          JSR    TSTCFL
143E  2D 03             BLT    LCOR7
1440  BD 14 A4          JSR    COMP
1443  BD 14 D5  LCOR7   JSR    DOWN
1446  CE 18 00          LDX    #COR5
1449  EB 09             ADDB   9,X
144B  A9 08             ADCA   8,X
144D  A7 26             STAA   $26,X
144F  E7 27             STAB   $27,X          Y DONE
1451  A6 2A             LDAA   $2A,X
1453  A7 0A             STAA   $A,X
1455  6A 05             DEC    5,X
1457  EE 02             LDX    2,X
1459  FF 18 24          STX    COR1
145C  FE 18 26          LDX    COR3
145F  FF 18 08          STX    COR15
1462  B6 18 05          LDAA   COR12
1465  26 8F             BNE    LCOR4
1467  CE 18 00          LDX    #COR5
146A  6D 00             TST    0,X
146C  27 03             BEQ    LCOR8
```

```
        146E  BD 14 BC             JSR     DOWN2
        1471  6D 01      LCOR8     TST     1,X
        1473  27 1B                BEQ     LCOR12
        1475  A6 24                LDAA    $24,X
        1477  A7 25                STAA    $25,X
        1479  2A 06                BPL     LCOR9
        147B  86 FF                LDAA    #$FF
        147D  A7 24                STAA    $24,X
        147F  20 02                BRA     LCOR10
        1481  6F 24      LCOR9     CLR     $24,X
        1483  A6 26      LCOR10    LDAA    $26,X
        1485  A7 27                STAA    $27,X
        1487  2A 05                BPL     LCOR11
        1489  86 FF                LDAA    #$FF
        148B  A7 26                STAA    $26,X
        148D  39                   RTS
        148E  6F 26      LCOR11    CLR     $26,X
        1490  39         LCOR12    RTS
                         *
                         *
                         *         RETURNS RESCALE FACTOR IN A
        1491  4D         SCALE     TSTA
        1492  27 07                BEQ     LSC3
        1494  43                   COMA
        1495  27 03                BEQ     LSC2
        1497  86 00      LSC1      LDAA    #0
        1499  39                   RTS
        149A  50         LSC2      NEGB
        149B  C4 80      LSC3      ANDB    #$80
        149D  27 02                BEQ     LSC4
        149F  20 F6                BRA     LSC1
        14A1  86 01      LSC4      LDAA    #1
        14A3  39                   RTS
                         *
                         *
                         *         RETURNS 2'S COMP IN A,B
        14A4  50         COMP      NEGB
        14A5  25 02                BCS     LCP1
        14A7  40                   NEGA
        14A8  39                   RTS
        14A9  43         LCP1      COMA
        14AA  39                   RTS
                         *
                         *
                         *         RETURNS CONDITION CODE REGISTOR STATE
        14AB  6D 0C      TSTCFL    TST     $C,X
        14AD  2E 0A                BGT     LT2
        14AF  6D 0A                TST     $A,X
        14B1  2C 03                BGE     LT1
        14B3  6D 0B                TST     $B,X
        14B5  39                   RTS
        14B6  6D 0C      LT1       TST     $C,X
        14B8  39                   RTS
        14B9  6D 26      LT2       TST     $26,X
        14BB  39                   RTS
                         *
                         *
```

```
14BC  A6 24   DOWN2   LDAA   $24,X
14BE  E6 25           LDAB   $25,X
14C0  47              ASRA
14C1  56              RORB
14C2  47              ASRA
14C3  56              RORB
14C4  A7 24           STAA   $24,X
14C6  E7 25           STAB   $25,X
14C8  A6 26           LDAA   $26,X
14CA  E6 27           LDAB   $27,X
14CC  47              ASRA
14CD  56              RORB

14CE  47              ASRA
14CF  56              RORB
14D0  A7 26           STAA   $26,X
14D2  E7 27           STAB   $27,X
14D4  39              RTS
                *
                *
                *     RETURNS A,B SCALED DOWN 2**-(I-2) BITS
14D5  EE 0D   DOWN    LDX    $D,X
14D7  6E 00           JMP    0,X
14D9  20 2E   DL0     BRA    DL8
14DB  39              RTS
14DC  39              RTS
14DD  39              RTS
14DE  39              RTS
14DF  20 26           BRA    DL7
14E1  20 22           BRA    DL6
14E3  20 1E           BRA    DL5
14E5  20 1A           BRA    DL4
14E7  20 16           BRA    DL3
14E9  20 12           BRA    DL2
14EB  20 0E           BRA    DL1
14ED  20 1B           BRA    DL9
14EF  20 1C           BRA    DL11
14F1  20 1E           BRA    DL12
14F3  20 20           BRA    DL13
14F5  20 22           BRA    DL14
14F7  20 24           BRA    DL15
14F9  20 26           BRA    DL16
14FB  47      DL1     ASRA
14FC  56              RORB
14FD  47      DL2     ASRA
14FE  56              RORB
14FF  47      DL3     ASRA
1500  56              RORB
1501  47      DL4     ASRA
1502  56              RORB
1503  47      DL5     ASRA
1504  56              RORB
1505  47      DL6     ASRA
1506  56              RORB
1507  47      DL7     ASRA
1508  56              RORB
1509  39      DL8     RTS
```

```
150A  8D 1E   DL9     BSR   SWITCH
150C  39      DL10    RTS
150D  8D 1B   DL11    BSR   SWITCH
150F  20 17           BRA   DL22
1511  8D 17   DL12    BSR   SWITCH
1513  20 12           BRA   DL21
1515  8D 13   DL13    BSR   SWITCH
1517  20 0D           BRA   DL20
1519  8D 0F   DL14    BSR   SWITCH
151B  20 08           BRA   DL19
151D  8D 0B   DL15    BSR   SWITCH
151F  20 03           BRA   DL18
1521  8D 07   DL16    BSR   SWITCH
1523  57              ASRB
1524  57      DL18    ASRB
1525  57      DL19    ASRB
1526  57      DL20    ASRB
1527  57      DL21    ASRB
1528  57      DL22    ASRB
1529  39              RTS
              *
152A  16      SWITCH  TAB
152B  2A 03           BPL   SW1
152D  86 FF           LDAA  #$FF
152F  39              RTS
1530  4F      SW1     CLRA
1531  39              RTS
              *
1532          CTABLE  EQU   *
1532  40 00           FCB   $40,0
1534  20 00           FCB   $20,0
1536  12 E4           FCB   $12,$E4
1538  09 FB           FCB   9,$FB
153A  05 11           FCB   5,$11
153C  02 8B           FCB   2,$8B
153E  01 46           FCB   1,$46
1540  00 A3           FCB   0,$A3
1542  00 51           FCB   0,$51
1544  00 29           FCB   0,$29
1546  00 14           FCB   0,$14
1548  00 0A           FCB   0,$A
154A  00 05           FCB   0,5
154C  00 03           FCB   0,3
154E  00 01           FCB   0,1
1550  00 01           FCB   0,1
1552                  END
```

STATEMENTS =762

FREE BYTES =16485

NO ERRORS DETECTED

```
1800                        ORG     $1800
1800             TEMP1      RMB     1
1801             TEMP2      RMB     1
1802             TEMP3      RMB     1
1803             TEMP4      RMB     1
1804             TEMP5      RMB     1
1805             TEMP6      RMB     1
1806             BCD1       RMB     1
1807             BCD2       RMB     1
1808             CTR        RMB     1
1809             CTR1       RMB     1
1824                        ORG     $1824
1824             COR1       RMB     1
1825             COR2       RMB     1
1826             COR3       RMB     1
1827             COR4       RMB     3
182A             COR9       RMB     1
182B             COR10      RMB     5
1830             UI1        RMB     1
1831             UI2        RMB     1
1832             VJ1        RMB     1
1833             VJ2        RMB     2
1835             DTIME      RMB     8
183D             NSAMP      RMB     1
183E             LOGS       RMB     16
184E             STACK1     RMB     1
184F             STACK2     RMB     1
1850             PUSHST     RMB     1
18A1                        ORG     $18A1
18A1             SIGNI      RMB     1
18A2             SIGNQ      RMB     1
18EC                        ORG     $18EC
18EC             I          RMB     1
18ED             P          RMB     2
18EF             THETA      RMB     4
18F3             PHI        RMB     2
18FF                        ORG     $18FF
18FF             INT        RMB     4
1903             S1         RMB     8
190B             S2         RMB     8
1913             S3         RMB     8
191B             AVG        RMB     6
8000             MATH       EQU     $8000
330F             SYNDET     EQU     $330F
33B8             MUXSEL     EQU     $33B8
1376             VECT15     EQU     $1376
2700                        ORG     $2700
                 *          SUBROUTINE RCORR
                 *          CORRELATION OF TWO REAL SEQUENCES
                 *          CLEAR S1,S2,S3 FLOAT PT NS.
                 *          ACCUMULATE IN EXTERNAL LOOP; ENTER AT RCORR1
                 *          X POINTS TO CURRENT SAMPLE
                 *          ENTER AT RCORR2 TO COMPUTE S1/(S2*S3)**0.5
2700  FF 18 00   RCORR1     STX     TEMP1
2703  A6 00                 LDAA    0,X
2705  E6 01                 LDAB    1,X
2707  BD 28 7E              JSR     PUSH82
```

```
270A  FE 18 00          LDX    TEMP1
270D  A6 02             LDAA   2,X
270F  E6 03             LDAB   3,X
2711  BD 28 7E          JSR    PUSH82
2714  86 01             LDAA   #1
2716  BD 2A EB          JSR    MATH1      X(I)*Y(I)
2719  CE 19 03          LDX    #S1
271C  BD 28 64          JSR    PUSH88
271F  86 06             LDAA   #6
2721  BD 80 00          JSR    MATH       S1=S1+X(I)*Y(I)
2724  CE 19 03          LDX    #S1
2727  BD 28 4A          JSR    PULL8
272A  FE 18 00          LDX    TEMP1
272D  A6 00             LDAA   0,X
272F  E6 01             LDAB   1,X
2731  BD 28 7E          JSR    PUSH82
2734  FE 18 00          LDX    TEMP1
2737  A6 00             LDAA   0,X
2739  E6 01             LDAB   1,X
273B  BD 28 7E          JSR    PUSH82
273E  86 01             LDAA   #1
2740  BD 2A EB          JSR    MATH1      X(I)*X(I)
2743  CE 19 0B          LDX    #S2
2746  BD 28 64          JSR    PUSH88
2749  86 06             LDAA   #6
274B  BD 80 00          JSR    MATH       S2=S2+X(I)*X(I)
274E  CE 19 0B          LDX    #S2
2751  BD 28 4A          JSR    PULL8
2754  FE 18 00          LDX    TEMP1
2757  A6 02             LDAA   2,X
2759  E6 03             LDAB   3,X
275B  BD 28 7E          JSR    PUSH82
275E  FE 18 00          LDX    TEMP1
2761  A6 02             LDAA   2,X
2763  E6 03             LDAB   3,X
2765  BD 28 7E          JSR    PUSH82
2768  86 01             LDAA   #1
276A  BD 2A EB          JSR    MATH1      Y(I)*Y(I)
276D  CE 19 13          LDX    #S3
2770  BD 28 64          JSR    PUSH88
2773  86 06             LDAA   #6
2775  BD 80 00          JSR    MATH       S3=Y(I)*Y(I)
2778  CE 19 13          LDX    #S3
277B  BD 28 4A          JSR    PULL8
277E  39                RTS
277F  CE 19 0B  RCORR2  LDX    #S2
2782  BD 28 64          JSR    PUSH88
2785  CE 19 13          LDX    #S3
2788  BD 28 64          JSR    PUSH88
278B  86 01             LDAA   #1
278D  BD 2A EB          JSR    MATH1      S2*S3
2790  CE 19 0B          LDX    #S2
2793  BD 28 4A          JSR    PULL8
2796  FE 19 0B          LDX    S2         COPY
2799  FF 19 13          STX    S3
279C  FE 19 0D          LDX    S2+2
279F  FF 19 15          STX    S3+2
```

```
27A2  FE 19 0F          LDX   S2+4
27A5  FF 19 17          STX   S3+4
27A8  FE 19 11          LDX   S2+6
27AB  FF 19 19          STX   S3+6
27AE  B6 19 1A          LDAA  S3+7       EXP
27B1  47                ASRA
27B2  B7 19 1A          STAA  S3+7       S3=INITIAL ITERATE
27B5  86 0A             LDAA  #10D
27B7  B7 18 00          STAA  TEMP1
27BA  CE 19 0B    RC1   LDX   #S2
27BD  BD 28 64          JSR   PUSH88
27C0  CE 19 13          LDX   #S3
27C3  BD 28 64          JSR   PUSH88
27C6  86 02             LDAA  #2
27C8  BD 80 00          JSR   MATH       X=S2/S3
27CB  CE 19 13          LDX   #S3
27CE  BD 28 64          JSR   PUSH88
27D1  86 06             LDAA  #6
27D3  BD 80 00          JSR   MATH       X=X+S3
27D6  30                TSX
27D7  6A 07             DEC   7,X        X=0.5X
27D9  CE 19 13          LDX   #S3
27DC  BD 28 4A          JSR   PULL8
27DF  7A 18 00          DEC   TEMP1
27E2  26 D6             BNE   RC1
27E4  CE 19 03          LDX   #S1        SQUARE ROOT DONE
27E7  BD 28 64          JSR   PUSH88
27EA  CE 19 13          LDX   #S3
27ED  BD 28 64          JSR   PUSH88
27F0  86 02             LDAA  #2
27F2  BD 80 00          JSR   MATH
27F5  CE 19 03          LDX   #S1
27F8  BD 28 4A          JSR   PULL8      S1=S1/(S2*S3)**0.5
27FB  F6 19 0A          LDAB  S1+7
27FE  2E 43             BGT   RC4
2800  50                NEGB
2801  CE 19 03          LDX   #S1
2804  5D          RC2   TSTB             FLOAT TO FIX
2805  27 17             BEQ   RC3
2807  A6 00             LDAA  0,X
2809  47                ASRA
280A  A7 00             STAA  0,X
280C  A6 01             LDAA  1,X
280E  46                RORA
280F  A7 01             STAA  1,X
2811  A6 02             LDAA  2,X
2813  46                RORA
2814  A7 02             STAA  2,X
2816  A6 03             LDAA  3,X
2818  46                RORA
2819  A7 03             STAA  3,X
281B  5A                DECB
281C  20 E6             BRA   RC2
281E  CE 19 03    RC3   LDX   #S1
2821  4F                CLRA
2822  BD 28 C9          JSR   PUSH44
2825  86 DC             LDAA  #$DC
```

```
2827  36               PSHA
2828  86 46            LDAA    #$46
282A  36               PSHA
282B  86 03            LDAA    #$03
282D  36               PSHA
282E  4F               CLRA
282F  36               PSHA                SF=65536*32768/10000
2830  86 09            LDAA    #9
2832  BD 80 00         JSR     MATH        SCALE FOR BCD
2835  CE 19 03         LDX     #S1
2838  4F               CLRA
2839  BD 28 A3         JSR     PULL4
283C  FE 19 05         LDX     S1+2
283F  FF 18 02         STX     TEMP3
2842  39               RTS
2843  CE 27 0F  RC4    LDX     #$270F      EXP>0
2846  FF 18 02         STX     TEMP3
2849  39               RTS
                *
                *      SUBROUTINE PULL8
                *      PULLS FLOATING PT N
                *      X POINTS TO MSB
284A  33        PULL8  PULB
284B  F7 18 4E         STAB    STACK1
284E  33               PULB
284F  F7 18 4F         STAB    STACK2
2852  C6 08            LDAB    #8
2854  32        P8     PULA
2855  A7 00            STAA    0,X
2857  08               INX
2858  5A               DECB
2859  26 F9            BNE     P8
285B  F6 18 4F         LDAB    STACK2
285E  37               PSHB
285F  F6 18 4E         LDAB    STACK1
2862  37               PSHB
2863  39               RTS
                *
                *      SUBROUTINE PUSH88
                *      PUSHES N FLOATING PT N ON STACK
                *      X POINTS TO MSB
2864  33        PUSH88 PULB
2865  F7 18 4E         STAB    STACK1
2868  33               PULB
2869  F7 18 4F         STAB    STACK2
286C  C6 08            LDAB    #8
286E  A6 07     P88    LDAA    7,X
2870  36               PSHA
2871  09               DEX
2872  5A               DECB
2873  26 F9            BNE     P88
2875  F6 18 4F         LDAB    STACK2
2878  37               PSHB
2879  F6 18 4E         LDAB    STACK1
287C  37               PSHB
287D  39               RTS
                *
```

```
                          *         SUBROUTINE PUSH82
                          *         PUSHES 2 BYTE NUMBER ON STACK
                          *         CONVERTS TO FLOAT
                          *         MSB IN A, LSB IN B
287E  F7 18 50    PUSH82    STAB    PUSHST
2881  33                    PULB
2882  F7 18 4E              STAB    STACK1
2885  33                    PULB
2886  F7 18 4F              STAB    STACK2
2889  5F                    CLRB
288A  37                    PSHB
288B  37                    PSHB
288C  37                    PSHB
288D  37                    PSHB
288E  37                    PSHB
288F  37                    PSHB
2890  F6 18 50              LDAB    PUSHST
2893  37                    PSHB
2894  36                    PSHA
2895  86 07                 LDAA    #7
2897  BD 80 00              JSR     MATH            NFPN
289A  F6 18 4F              LDAB    STACK2
289D  37                    PSHB
289E  F6 18 4E              LDAB    STACK1
28A1  37                    PSHB
28A2  39                    RTS
                          *
                          *         SUBROUTINE PULL4
                          *         PULLS 4 BYTES OFF STACK
                          *         STORES MSB IN X+A
28A3  33          PULL4     PULB
28A4  F7 18 4E              STAB    STACK1
28A7  33                    PULB
28A8  F7 18 4F              STAB    STACK2
28AB  4D                    TSTA
28AC  27 06                 BEQ     LP42
28AE  08          LP41      INX
28AF  4A                    DECA
28B0  27 02                 BEQ     LP42
28B2  20 FA                 BRA     LP41
28B4  32          LP42      PULA
28B5  A7 00                 STAA    0,X
28B7  32                    PULA
28B8  A7 01                 STAA    1,X
28BA  32                    PULA
28BB  A7 02                 STAA    2,X
28BD  32                    PULA
28BE  A7 03                 STAA    3,X
28C0  F6 18 4F              LDAB    STACK2
28C3  37                    PSHB
28C4  F6 18 4E              LDAB    STACK1
28C7  37                    PSHB
28C8  39                    RTS
                          *
                          *         SUBROUTINE PUSH44
                          *         PUSHES 4 BYTE NUMBER ONTO STACK
                          *         MSB IN X+A
```

```
28C9   33           PUSH44   PULB
28CA   F7 18 4E              STAB     STACK1
28CD   33                    PULB
28CE   F7 18 4F              STAB     STACK2
28D1   4D                    TSTA
28D2   27 06                 BEQ      LP442
28D4   08           LP441    INX
28D5   4A                    DECA
28D6   27 02                 BEQ      LP442
28D8   20 FA                 BRA      LP441
28DA   A6 03        LP442    LDAA     3,X
28DC   36                    PSHA
28DD   A6 02                 LDAA     2,X
28DF   36                    PSHA
28E0   A6 01                 LDAA     1,X
28E2   36                    PSHA
28E3   A6 00                 LDAA     0,X
28E5   36                    PSHA
28E6   F6 18 4F              LDAB     STACK2
28E9   37                    PSHB
28EA   F6 18 4E              LDAB     STACK1
28ED   37                    PSHB
28EE   39                    RTS
                    *
                    *        SUBROUTINE PUSH41
                    *        PUSHES 4 BYTE NUMBER ONTO STACK
                    *        3 MS BYTES ARE SIGN BITS
                    *        ENTER WITH LS BYTE IN A REG
28EF   33           PUSH41   PULB
28F0   F7 18 4E              STAB     STACK1
28F3   33                    PULB
28F4   F7 18 4F              STAB     STACK2
28F7   36                    PSHA
28F8   48                    ASLA
28F9   24 04                 BCC      LP411
28FB   86 FF                 LDAA     #$FF              NEGATIVE
28FD   20 01                 BRA      LP412
28FF   4F           LP411    CLRA                       POSITIVE
2900   36           LP412    PSHA
2901   36                    PSHA
2902   36                    PSHA
2903   F6 18 4F              LDAB     STACK2
2906   37                    PSHB
2907   F6 18 4E              LDAB     STACK1
290A   37                    PSHB
290B   39                    RTS
                    *
                    *        SUBROUTINE PUSH42
                    *        PUSHES 4 BYTE NUMBER ONTO STACK
                    *        2 MS BYTER ARE SIGN BITS
                    *        ENTER WITH LS BYTE IN B, NEXT LS BYTER IN A
290C   F7 18 50     PUSH42   STAB     PUSHST
290F   33                    PULB
2910   F7 18 4E              STAB     STACK1
2913   33                    PULB
2914   F7 18 4F              STAB     STACK2
2917   F6 18 50              LDAB     PUSHST
```

```
291A  37                      PSHB
291B  36                      PSHA
291C  48                      ASLA
291D  24 04                   BCC     LP421
291F  86 FF                   LDAA    #$FF           NEGATIVE
2921  20 01                   BRA     LP422
2923  4F          LP421       CLRA                   POSITIVE
2924  36          LP422       PSHA
2925  36                      PSHA
2926  F6 18 4F                LDAB    STACK2
2929  37                      PSHB
292A  F6 18 4E                LDAB    STACK1
292D  37                      PSHB
292E  39                      RTS
                  *
                  *           SUBROUTINE TRAPIN
                  *           TRAPEZOIDAL INTEGRATION
292F  F6 18 ED    TRAPIN      LDAB    P
2932  B6 19 02    T1          LDAA    INT+3
2935  BB 18 25                ADDA    COR2
2938  B7 19 02                STAA    INT+3
293B  B6 19 01                LDAA    INT+2
293E  B9 18 24                ADCA    COR1
2941  B7 19 01                STAA    INT+2
2944  B6 19 00                LDAA    INT+1
2947  89 00                   ADCA    #0
2949  B7 19 00                STAA    INT+1
294C  B6 18 FF                LDAA    INT
294F  89 00                   ADCA    #0
2951  B7 18 FF                STAA    INT
2954  5D                      TSTB
2955  27 03                   BEQ     T2
2957  5A                      DECB
2958  20 D8                   BRA     T1
295A  86 01       T2          LDAA    #1
295C  B7 18 ED                STAA    P
295F  39                      RTS
                  *
                  *           SUBROUTINE TRAPFX
                  *           FINAL SAMPLE: TRAP INT
2960  86 02       TRAPFX      LDAA    #2
2962  BD 29 99                JSR     DELAY1
2965  BD 2A 4C                JSR     RDDEFT
2968  BD 13 76                JSR     VECT15
296B  7F 18 ED                CLR     P
296E  BD 29 2F                JSR     TRAPIN
2971  CE 18 FF                LDX     #INT
2974  A6 21                   LDAA    $21,X
2976  AB 03                   ADDA    3,X
2978  A7 21                   STAA    $21,X
297A  A6 20                   LDAA    $20,X
297C  A9 02                   ADCA    2,X
297E  A7 20                   STAA    $20,X
2980  A6 1F                   LDAA    $1F,X
2982  A9 01                   ADCA    1,X
2984  A7 1F                   STAA    $1F,X
2986  A6 1E                   LDAA    $1E,X
```

```
2988  A9 00                 ADCA    0,X
298A  A7 1E                 STAA    $1E,X
298C  A6 1D                 LDAA    $1D,X
298E  89 00                 ADCA    #0
2990  A7 1D                 STAA    $1D,X
2992  A6 1C                 LDAA    $1C,X
2994  89 00                 ADCA    #0
2996  A7 1C                 STAA    $1C,X
2998  39                    RTS
                    *
                    *       SUBROUTINE DELAY1
                    *       9.996MS*DTIME DELAY
2999  B6 18 35      DELAY1  LDAA    DTIME
299C  4D            D1      TSTA
299D  27 05                 BEQ     D2
299F  8D 04                 BSR     DELAY
29A1  4A                    DECA
29A2  20 F8                 BRA     D1
29A4  39            D2      RTS
                    *
                    *       SUBROUTINE DELAY
29A5  CE 02 CA      DELAY   LDX     #$2CA
29A8  09            LDY1    DEX
29A9  27 02                 BEQ     LDY2
29AB  20 FB                 BRA     LDY1
29AD  39            LDY2    RTS
                    *
                    *       SUBROUTINE BI$BCD
                    *       POSITIVE BINARY IN TEMP3,TEMP4
                    *       RETURNS BCD IN BCD1,BCD2
29AE  7F 18 06      BI$BCD  CLR     BCD1
29B1  7F 18 07              CLR     BCD2
29B4  86 10                 LDAA    #16D
29B6  B7 18 09              STAA    CTR1
29B9  BF 18 04              STS     TEMP5
29BC  8E 18 01              LDS     #TEMP3-1
29BF  86 08         LBIB1   LDAA    #8D
29C1  B7 18 08              STAA    CTR
29C4  33                    PULB
29C5  58            LBIB2   ASLB
29C6  24 12                 BCC     LBIB3
29C8  86 01                 LDAA    #1
29CA  BB 18 07              ADDA    BCD2
29CD  19                    DAA
29CE  B7 18 07              STAA    BCD2
29D1  B6 18 06              LDAA    BCD1
29D4  89 00                 ADCA    #0
29D6  19                    DAA
29D7  B7 18 06              STAA    BCD1
29DA  7A 18 09      LBIB3   DEC     CTR1
29DD  27 1B                 BEQ     LBIB4
29DF  B6 18 07              LDAA    BCD2
29E2  BB 18 07              ADDA    BCD2
29E5  19                    DAA
29E6  B7 18 07              STAA    BCD2
29E9  B6 18 06              LDAA    BCD1
29EC  B9 18 06              ADCA    BCD1
```

```
29EF  19                  DAA
29F0  B7 18 06            STAA    BCD1
29F3  7A 18 08            DEC     CTR
29F6  27 C7               BEQ     LBIB1
29F8  20 CB               BRA     LBIB2
29FA  BE 18 04   LBIB4    LDS     TEMP5
29FD  39                  RTS
                 *
                 *        SUBROUTINE TUNE
29FE  B6 18 30   TUNE     LDAA    UI1
2A01  B7 EE 21            STAA    $EE21
2A04  B6 18 31            LDAA    UI2
2A07  B7 EE 20            STAA    $EE20
2A0A  B6 18 32            LDAA    VJ1
2A0D  B7 EE 11            STAA    $EE11
2A10  B6 18 33            LDAA    VJ2
2A13  B7 EE 10            STAA    $EE10
2A16  39                  RTS
                 *
                 *        SUBROUTINES IDEX
2A17  FF 18 00   IDEX     STX     TEMP1
2A1A  FB 18 01            ADDB    TEMP2
2A1D  B9 18 00            ADCA    TEMP1
2A20  B7 18 00            STAA    TEMP1
2A23  F7 18 01            STAB    TEMP2
2A26  FE 18 00            LDX     TEMP1
2A29  39                  RTS
                 *
                 *        SUBROUTINE WRITE
2A2A  A6 00      WRITE    LDAA    0,X
2A2C  44                  LSRA
2A2D  44                  LSRA
2A2E  44                  LSRA
2A2F  44                  LSRA
2A30  8A 30               ORAA    #$30
2A32  BD FD 80            JSR     $FD80
2A35  A6 00               LDAA    0,X
2A37  84 0F               ANDA    #$0F
2A39  8A 30               ORAA    #$30
2A3B  BD FD 80            JSR     $FD80
2A3E  39                  RTS
                 *
                 *        SUBROUTINE PASC
2A3F  5D         PASC     TSTB
2A40  27 09               BEQ     PAS
2A42  A6 00               LDAA    0,X
2A44  BD FD 80            JSR     $FD80
2A47  08                  INX
2A48  5A                  DECB
2A49  20 F4               BRA     PASC
2A4B  39         PAS      RTS
                 *
                 *        SUBROUTINE RDDEFT
2A4C  B6 18 3D   RDDEFT   LDAA    NSAMP
2A4F  B7 18 08            STAA    CTR
2A52  CE 00 00            LDX     #0
2A55  FF 18 24            STX     COR1
```

```
2A58  FF 18 26          STX    COR3
2A5B  FF 18 28          STX    COR4+1
2A5E  7D 18 08   RD1    TST    CTR
2A61  27 2D             BEQ    RD2
2A63  8D 63             BSR    READSN
2A65  CE 18 00          LDX    #$1800
2A68  A6 26             LDAA   $26,X
2A6A  AB 01             ADDA   1,X
2A6C  A7 26             STAA   $26,X
2A6E  A6 25             LDAA   $25,X
2A70  A9 00             ADCA   0,X
2A72  A7 25             STAA   $25,X
2A74  A6 24             LDAA   $24,X
2A76  A9 A1             ADCA   $A1,X
2A78  A7 24             STAA   $24,X
2A7A  A6 29             LDAA   $29,X
2A7C  AB 03             ADDA   3,X
2A7E  A7 29             STAA   $29,X
2A80  A6 28             LDAA   $28,X
2A82  A9 02             ADCA   2,X
2A84  A7 28             STAA   $28,X
2A86  A6 27             LDAA   $27,X
2A88  A9 A2             ADCA   $A2,X
2A8A  A7 27             STAA   $27,X
2A8C  6A 08             DEC    8,X
2A8E  20 CE             BRA    RD1
2A90  A6 3E      RD2    LDAA   $3E,X
2A92  A7 08             STAA   8,X
2A94  6D 08      RD3    TST    8,X
2A96  27 23             BEQ    RD4
2A98  A6 24             LDAA   $24,X
2A9A  47                ASRA
2A9B  A7 24             STAA   $24,X
2A9D  A6 25             LDAA   $25,X
2A9F  46                RORA
2AA0  A7 25             STAA   $25,X
2AA2  A6 26             LDAA   $26,X
2AA4  46                RORA
2AA5  A7 26             STAA   $26,X
2AA7  A6 27             LDAA   $27,X
2AA9  47                ASRA
2AAA  A7 27             STAA   $27,X
2AAC  A6 28             LDAA   $28,X
2AAE  46                RORA
2AAF  A7 28             STAA   $28,X
2AB1  A6 29             LDAA   $29,X
2AB3  46                RORA
2AB4  A7 29             STAA   $29,X
2AB6  7A 18 08          DEC    CTR
2AB9  20 D9             BRA    RD3
2ABB  FE 18 25   RD4    LDX    COR2
2ABE  FF 18 24          STX    COR1
2AC1  FE 18 28          LDX    COR4+1
2AC4  FF 18 26          STX    COR3
2AC7  39                RTS
                 *
                 *      SUBROUTINE READSN
```

```
2AC8  86 03     READSN  LDAA   #3
2ACA  B7 33 B8          STAA   MUXSEL
2ACD  CE 18 00          LDX    #TEMP1
2AD0  BD 33 0F          JSR    SYNDET
2AD3  7D 18 00          TST    TEMP1
2AD6  8D 0C             BSR    SGNSET
2AD8  B7 18 A1          STAA   SIGNI
2ADB  7D 18 02          TST    TEMP3
2ADE  8D 04             BSR    SGNSET
2AE0  B7 18 A2          STAA   SIGNQ
2AE3  39                RTS
                *
2AE4  2A 03     SGNSET  BPL    SETPOS
2AE6  86 FF             LDAA   #$FF
2AE8  39                RTS
2AE9  4F        SETPOS  CLRA
2AEA  39                RTS
                *
2AEB  33        MATH1   PULB
2AEC  F7 18 4E          STAB   STACK1
2AEF  33                PULB
2AF0  F7 18 4F          STAB   STACK2
2AF3  30                TSX
2AF4  E6 0F             LDAB   15D,X
2AF6  C1 80             CMPB   #$80
2AF8  27 05             BEQ    M11
2AFA  BD 80 00          JSR    MATH
2AFD  20 06             BRA    M13
2AFF  86 08     M11     LDAA   #8
2B01  33        M12     PULB
2B02  4A                DECA
2B03  26 FC             BNE    M12
2B05  F6 18 4F  M13     LDAB   STACK2
2B08  37                PSHB
2B09  F6 18 4E          LDAB   STACK1
2B0C  37                PSHB
2B0D  39                RTS
                *
2B0E  BD FD A6  PRINT   JSR    $FDA6
2B11  CE 18 EC          LDX    #I
2B14  C6 01             LDAB   #1
2B16  BD FD 74          JSR    $FD74
2B19  86 20             LDAA   #$20
2B1B  BD FD 80          JSR    $FD80
2B1E  CE 18 30          LDX    #UI1
2B21  BD 2A 2A          JSR    WRITE
2B24  CE 18 31          LDX    #UI2
2B27  BD 2A 2A          JSR    WRITE
2B2A  86 20             LDAA   #$20
2B2C  BD FD 80          JSR    $FD80
2B2F  CE 18 32          LDX    #VJ1
2B32  BD 2A 2A          JSR    WRITE
2B35  CE 18 33          LDX    #VJ2
2B38  BD 2A 2A          JSR    WRITE
2B3B  86 20             LDAA   #$20
2B3D  BD FD 80          JSR    $FD80
2B40  CE 18 EF          LDX    #THETA
```

```
2B43  C6 02          LDAB    #2
2B45  BD FD 74       JSR     $FD74
2B48  86 20          LDAA    #$20
2B4A  BD FD 80       JSR     $FD80
2B4D  CE 18 F3       LDX     #PHI
2B50  C6 02          LDAB    #2
2B52  BD FD 74       JSR     $FD74
2B55  39             RTS
2B56                 END
```

STATEMENTS =637

FREE BYTES =17972

NO ERRORS DETECTED

ASSEMBLE

APPENDIX B - Method of Invariant Moments Assembly Code Listing

This Appendix consists of a listing of the assembly language program which computes invariant Fourier signatures. This program was written to run on the Deft Laboratories' microprocessor-based test bed. All addresses and opcodes are hexadecimal. In the operand column of the statements the following symbols are used:

$	Hexadecimal Prefix
%	Binary Prefix
H	Hexadecimal Postfix
D	Decimal Postfix
B	Binary Postfix
#	Denotes Immediate Addressing Mode

The entry address for this program is $2000.

METHOD OF INVARIANT MOMENTS

```
1802                    ORG     $1802
1802            TEMP3   RMB     1
1803            TEMP4   RMB     3
1806            BCD1    RMB     1
1807            BCD2    RMB     1
182A                    ORG     $182A
182A            COR9    RMB     1
182B            COR10   RMB     5
1830            UI1     RMB     1
1831            UI2     RMB     1
1832            VJ1     RMB     1
1833            VJ2     RMB     2
1835            DTIME   RMB     1
1836            I       RMB     1
1837            J       RMB     1
1838            K       RMB     1
1839            XST     RMB     2
183B            VVJ1    RMB     1
183C            VVJ2    RMB     1
183D            NSAMP   RMB     1
183E            LOGS    RMB     1
183F            DX      RMB     2
1841            DY      RMB     2
1843            DDY     RMB     2
1845            PO      RMB     2
1847            PX      RMB     2
1849            PY      RMB     2
184B            SIGN    RMB     3
184E            STACK1  RMB     1
184F            STACK2  RMB     1
1850            PUSHST  RMB     1
1851            LIMIT   RMB     1
18E0                    ORG     $18E0
18E0            IPX     RMB     2
18E2            IPY     RMB     2
18E4            X       RMB     8
18EC            Y       RMB     8
1B50                    ORG     $1B50
1B50            MOO     RMB     $200
8000            MATH    EQU     $8000
29EC            MATH1   EQU     $29EC
2AF0            BI$BCD  EQU     $2AF0
2800            FL$BCD  EQU     $2800
281F            FL2     EQU     $281F
2961            SAMP    EQU     $2961
2976            SAMPLE  EQU     $2976
2A0F            FZERO   EQU     $2A0F
29D7            INCX    EQU     $29D7
2872            ACCUM   EQU     $2872
2B5A            PUSH88  EQU     $2B5A
2B40            PULL8   EQU     $2B40
2A2A            SQROOT  EQU     $2A2A
2C17            PASC    EQU     $2C17
2C02            WRITE   EQU     $2C02
2CC2            C01     EQU     $2CC2
2D12            C23     EQU     $2D12
[illegible]     C45     EQU     $2D62
```

METHOD OF INVARIANT MOMENTS

```
2DB2                C67     EQU     $2DB2
2000                        ORG     $2000
2000  BD FD A6      CALIB   JSR     $FDA6
2003  CE 24 97              LDX     #LINE1
2006  C6 24                 LDAB    #36D
2008  BD 2C 17              JSR     PASC
200B  BD FD 36      CB1     JSR     $FD36           INCH
200E  81 43                 CMPA    #$43
2010  26 F9                 BNE     CB1
2012  FE 25 77              LDX     XZERO           MEASURE PO,PX,PY
2015  FF 18 30              STX     UI1
2018  FE 25 79              LDX     YZERO
201B  FF 18 32              STX     VJ1
201E  86 80                 LDAA    #$80
2020  B7 18 3D              STAA    NSAMP
2023  86 07                 LDAA    #7
2025  B7 18 3E              STAA    LOGS
2028  86 02                 LDAA    #2
202A  B7 18 35              STAA    DTIME
202D  BD 29 61              JSR     SAMP
2030  B7 18 45              STAA    PO
2033  F7 18 46              STAB    PO+1
2036  CE 99 93              LDX     #$9993
2039  FF 18 3F              STX     DX
203C  CE 00 07              LDX     #7
203F  FF 18 41              STX     DY
2042  BD 29 D7              JSR     INCX
2045  BD 29 61              JSR     SAMP
2048  F0 18 46              SUBB    PO+1
204B  B2 18 45              SBCA    PO
204E  B7 18 47              STAA    PX
2051  F7 18 48              STAB    PX+1
2054  FE 25 77              LDX     XZERO
2057  FF 18 30              STX     UI1
205A  BD 24 48              JSR     INCY
205D  BD 29 61              JSR     SAMP
2060  F0 18 46              SUBB    PO+1
2063  B2 18 45              SBCA    PO
2066  B7 18 49              STAA    PY
2069  F7 18 4A              STAB    PY+1
206C  BD FD A6      FEAT#2  JSR     $FDA6           ENTER TO SKIP CALIBRATIO
206F  CE 24 BB              LDX     #LINE2
2072  C6 26                 LDAB    #38D
2074  BD 2C 17              JSR     PASC
2077  BD FD A6              JSR     $FDA6
207A  BD FD A6      FE1     JSR     $FDA6
207D  CE 24 E1              LDX     #LINE3
2080  C6 15                 LDAB    #21D
2082  BD 2C 17              JSR     PASC
2085  BD FD 36              JSR     $FD36
2088  84 0F                 ANDA    #$0F
208A  B7 18 3E              STAA    LOGS
208D  C6 01                 LDAB    #1
208F  4A            FE2     DECA
2090  2D 03                 BLT     FE3
2092  58                    ASLB
2093  20 FA                 BRA     FE2
```

METHOD OF INVARIANT MOMENTS

```
2095  F7 18 3D  FE3    STAB   NSAMP
2098  BD FD A6         JSR    $FDA6
209B  CE 25 71         LDX    #LINE11
209E  C6 06            LDAB   #6
20A0  BD 2C 17         JSR    PASC
20A3  BD FD 36         JSR    $FD36
20A6  84 0F            ANDA   #$0F
20A8  B7 18 51         STAA   LIMIT

20AB  BD FD A6  FE3A   JSR    $FDA6
20AE  CE 24 F6         LDX    #LINE4
20B1  C6 21            LDAB   #33D
20B3  BD 2C 17         JSR    PASC
20B6  BD FD 36  FE4    JSR    $FD36
20B9  81 4D            CMPA   #$4D
20BB  26 F9            BNE    FE4
20BD  FE 25 79         LDX    YZERO
20C0  FF 18 32         STX    VJ1
20C3  FF 18 3B         STX    VVJ1
20C6  CE 99 93         LDX    #$9993
20C9  FF 18 3F         STX    DX
20CC  FF 18 43         STX    DDY
20CF  CE 00 07         LDX    #$7
20D2  FF 18 41         STX    DY
20D5  86 02            LDAA   #2
20D7  B7 18 35         STAA   DTIME
20DA  CE 00 00         LDX    #0
20DD  FF 18 E2         STX    IPY
20E0  CE 1B 50         LDX    #M00
20E3  86 40            LDAA   #64
20E5  BD 2A 0F  FE5    JSR    FZERO          INITIALIZE M(P,Q)
20E8  4A               DECA
20E9  26 FA            BNE    FE5
20EB  7F 18 37         CLR    J
20EE  B6 18 37  FE6    LDAA   J              WY LOOP
20F1  27 04            BEQ    FE7
20F3  84 03            ANDA   #3
20F5  27 4F            BEQ    FE13A
20F7  FE 25 77  FE7    LDX    XZERO          INITIALIZE WX LOOP
20FA  FF 18 30         STX    UI1
20FD  CE 00 00         LDX    #0
2100  FF 18 E0         STX    IPX
2103  7F 18 36         CLR    I
2106  B6 18 36  FE8    LDAA   I              WX LOOP
2109  27 07            BEQ    FE9
210B  84 03            ANDA   #3
210D  27 39            BEQ    FE13B

210F  44               LSRA
2110  25 0D            BCS    FE10
2112  BD 29 76  FE9    JSR    SAMPLE         X**P EVEN
2115  B6 18 37         LDAA   J
2118  27 12            BEQ    FE11
211A  44               LSRA
211B  25 31            BCS    FE14
211D  20 0D            BRA    FE11
211F  BD 29 76  FE10   JSR    SAMPLE         X**P ODD
```

```
2122  B6 18 37          LDAA   J
2125  27 43             BEQ    FE17
2127  44                LSRA
2128  25 5D             BCS    FE20
212A  20 3E             BRA    FE17
212C  CE 1B 50  FE11    LDX    #M00          X**P EVEN, Y**Q EVEN
212F  7F 18 38          CLR    K
2132  B6 18 38  FE12    LDAA   K
2135  5F                CLRB
2136  84 09             ANDA   #9
2138  26 01             BNE    FE13
213A  5C                INCB
213B  4F        FE13    CLRA
213C  BD 28 72          JSR    ACCUM
213F  8C 1D 50          CPX    #M00+512D
2142  26 EE             BNE    FE12
2144  20 5C             BRA    FE23
2146  20 6A     FE13A   BRA    FE24
2148  20 58     FE13B   BRA    FE23
214A  20 A2     FE13C   BRA    FE6
214C  20 B8     FE13D   BRA    FE8
214E  CE 1B 50  FE14    LDX    #M00          X**P EVEN, Y**Q ODD
2151  7F 18 38          CLR    K
2154  B6 18 38  FE15    LDAA   K
2157  84 09             ANDA   #9
2159  5F                CLRB
215A  4A                DECA
215B  26 01             BNE    FE16
215D  5C                INCB
215E  86 01     FE16    LDAA   #1
2160  BD 28 72          JSR    ACCUM
2163  8C 1D 50          CPX    #M00+512D
2166  26 EC             BNE    FE15
2168  20 38             BRA    FE23
216A  CE 1B 50  FE17    LDX    #M00          X**P ODD, Y**Q EVEN
216D  7F 18 38          CLR    K
2170  B6 18 38  FE18    LDAA   K
2173  84 09             ANDA   #9
2175  5F                CLRB
2176  81 08             CMPA   #8
2178  26 01             BNE    FE19
217A  5C                INCB
217B  86 02     FE19    LDAA   #2
217D  BD 28 72          JSR    ACCUM
2180  8C 1D 50          CPX    #M00+512D
2183  26 EB             BNE    FE18
2185  20 1B             BRA    FE23
2187  CE 1B 50  FE20    LDX    #M00          X**P ODD, Y**Q ODD
218A  7F 18 38          CLR    K
218D  B6 18 38  FE21    LDAA   K
2190  84 09             ANDA   #9
2192  5F                CLRB
2193  81 09             CMPA   #9
2195  26 01             BNE    FE22
2197  5C                INCB
2198  86 03     FE22    LDAA   #3
219A  BD 28 72          JSR    ACCUM
```

```
219D  8C 1D 50          CPX   #M00+512D
21A0  26 EB             BNE   FE21
21A2  7C 18 36   FE23   INC   I
21A5  BD 24 71          JSR   INCXP
21A8  BD 29 D7          JSR   INCX
21AB  B6 18 36          LDAA  I
21AE  81 28             CMPA  #40D
21B0  26 9A             BNE   FE13D
21B2  7C 18 37   FE24   INC   J
21B5  BD 24 84          JSR   INCYP
21B8  BD 24 48          JSR   INCY
21BB  B6 18 37          LDAA  J
21BE  81 28             CMPA  #40D
21C0  26 88             BNE   FE13C
21C2  CE 1B 58          LDX   #M00+8        NORMALIZE MOMENTS
21C5  FF 18 39          STX   XST
21C8  FE 18 39   FE25   LDX   XST
21CB  BD 2B 5A          JSR   PUSH88
21CE  CE 1B 50          LDX   #M00
21D1  BD 2B 5A          JSR   PUSH88
21D4  86 02             LDAA  #2
21D6  BD 80 00          JSR   MATH          M(P,Q)/M(0,0)
21D9  FE 18 39          LDX   XST
21DC  BD 2B 40          JSR   PULL8
21DF  FF 18 39          STX   XST
21E2  8C 1D 50          CPX   #M00+512D
21E5  26 E1             BNE   FE25
21E7  CE 1B 60          LDX   #M00+16D      M(0,2)
21EA  BD 2B 5A          JSR   PUSH88
21ED  CE 1B D0          LDX   #M00+128D     M(2,0)
21F0  BD 2B 5A          JSR   PUSH88
21F3  86 06             LDAA  #6
21F5  BD 80 00          JSR   MATH
21F8  CE 1B 58          LDX   #M00+8        M(0,1)
21FB  BD 2B 5A          JSR   PUSH88
21FE  CE 1B 58          LDX   #M00+8
2201  BD 2B 5A          JSR   PUSH88
2204  86 01             LDAA  #1
2206  BD 29 EC          JSR   MATH1
2209  86 05             LDAA  #5
220B  BD 80 00          JSR   MATH
220E  CE 1B 90          LDX   #M00+64D      M(1,0)
2211  BD 2B 5A          JSR   PUSH88
2214  CE 1B 90          LDX   #M00+64D
2217  BD 2B 5A          JSR   PUSH88
221A  86 01             LDAA  #1
221C  BD 29 EC          JSR   MATH1
221F  86 05             LDAA  #5
2221  BD 80 00          JSR   MATH
2224  CE 1B 50          LDX   #M00
2227  BD 2B 5A          JSR   PUSH88
222A  86 02             LDAA  #2
222C  BD 80 00          JSR   MATH
222F  CE 18 E4          LDX   #X
2232  BD 2B 40          JSR   PULL8         X MOMENT INVARIANT
2235  CE 1B 98          LDX   #M00+72D      M(1,1)
2238  BD 2B 5A          JSR   PUSH88
```

```
223B  CE 1B 58    LDX   #M00+8      M(0,1)
223E  BD 2B 5A    JSR   PUSH88
2241  CE 1B 90    LDX   #M00+64D    M(1,0)
2244  BD 2B 5A    JSR   PUSH88
2247  86 01       LDAA  #1
2249  BD 29 EC    JSR   MATH1
224C  86 05       LDAA  #5
224E  BD 80 00    JSR   MATH
2251  CE 18 EC    LDX   #Y
2254  BD 2B 40    JSR   PULL8
2257  CE 18 EC    LDX   #Y
225A  BD 2B 5A    JSR   PUSH88
225D  CE 18 EC    LDX   #Y
2260  BD 2B 5A    JSR   PUSH88
2263  86 01       LDAA  #1
2265  BD 29 EC    JSR   MATH1
2268  30          TSX
2269  6C 07       INC   7,X
226B  6C 07       INC   7,X
226D  CE 1B 58    LDX   #M00+8      M(0,1)
2270  BD 2B 5A    JSR   PUSH88
2273  CE 1B 58    LDX   #M00+8
2276  BD 2B 5A    JSR   PUSH88
2279  86 01       LDAA  #1
227B  BD 29 EC    JSR   MATH1
227E  CE 1B 90    LDX   #M00+64D    M(1,0)
2281  BD 2B 5A    JSR   PUSH88
2284  CE 1B 90    LDX   #M00+64D
2287  BD 2B 5A    JSR   PUSH88
228A  86 01       LDAA  #1
228C  BD 29 EC    JSR   MATH1
228F  86 05       LDAA  #5
2291  BD 80 00    JSR   MATH
2294  CE 1B D0    LDX   #M00+128D   M(2,0)
2297  BD 2B 5A    JSR   PUSH88
229A  86 06       LDAA  #6
229C  BD 80 00    JSR   MATH
229F  CE 1B 60    LDX   #M00+16D    M(0,2)
22A2  BD 2B 5A    JSR   PUSH88
22A5  86 05       LDAA  #5
22A7  BD 80 00    JSR   MATH
22AA  CE 18 EC    LDX   #Y
22AD  BD 2B 40    JSR   PULL8
22B0  CE 18 EC    LDX   #Y
22B3  BD 2B 5A    JSR   PUSH88
22B6  CE 18 EC    LDX   #Y
22B9  BD 2B 5A    JSR   PUSH88
22BC  86 01       LDAA  #1
22BE  BD 29 EC    JSR   MATH1
22C1  86 06       LDAA  #6
22C3  BD 80 00    JSR   MATH        Y**2
22C6  BD 2A 2A    JSR   SQROOT
22C9  CE 1B 50    LDX   #M00
22CC  BD 2B 5A    JSR   PUSH88
22CF  86 02       LDAA  #2
22D1  BD 80 00    JSR   MATH
22D4  CE 18 EC    LDX   #Y
```

```
22D7  BD 2B 40          JSR   PULL8      Y MOMENT INVARIANT
22DA  CE 1B 50          LDX   #M00
22DD  86 40             LDAA  #$40
22DF  A7 00             STAA  0,X
22E1  6F 01             CLR   1,X
22E3  6F 02             CLR   2,X
22E5  6F 03             CLR   3,X
22E7  6F 04             CLR   4,X
22E9  6F 05             CLR   5,X
22EB  6F 06             CLR   6,X
22ED  86 01             LDAA  #1
22EF  A7 07             STAA  7,X
22F1  BD FD A6          JSR   $FDA6
22F4  CE 25 17          LDX   #LINE5     PRINT MOMENTS
22F7  C6 06             LDAB  #6
22F9  BD 2C 17          JSR   PASC
22FC  BD FD A6          JSR   $FDA6
22FF  CE 25 1D          LDX   #LINE6
2302  C6 3E             LDAB  #62D
2304  BD 2C 17          JSR   PASC
2307  7F 18 36          CLR   I
230A  CE 1B 50          LDX   #M00
230D  FF 18 39          STX   XST
2310  BD FD A6    FE26  JSR   $FDA6
2313  B6 18 36          LDAA  I
2316  8A 30             ORAA  #$30
2318  BD FD 80          JSR   $FD80
231B  7F 18 37          CLR   J
231E  B6 18 37    FE27  LDAA  J
2321  B1 18 51          CMPA  LIMIT
2324  2A 25             BPL   FE28
2326  CE 25 1D          LDX   #LINE6
2329  C6 02             LDAB  #2
232B  BD 2C 17          JSR   PASC
232E  FE 18 39          LDX   XST
2331  BD 28 00          JSR   FL#BCD
2334  B6 18 4B          LDAA  SIGN
2337  BD FD 80          JSR   $FD80
233A  86 2E             LDAA  #$2E
233C  BD FD 80          JSR   $FD80
233F  CE 18 06          LDX   #BCD1
2342  BD 2C 02          JSR   WRITE
2345  CE 18 07          LDX   #BCD2
2348  BD 2C 02          JSR   WRITE
234B  FE 18 39    FE28  LDX   XST
234E  08                INX
234F  08                INX
2350  08                INX
2351  08                INX
2352  08                INX
2353  08                INX
2354  08                INX
2355  08                INX
2356  FF 18 39          STX   XST
2359  7C 18 37          INC   J
235C  B6 18 37          LDAA  J
235F  81 08             CMPA  #8
```

```
2361  26 BB             BNE    FE27
2363  7C 18 36          INC    I
2366  B6 18 36          LDAA   I
2369  B1 18 51          CMPA   LIMIT
236C  2B A2             BMI    FE26
236E  BD FD A6          JSR    $FDA6
2371  BD FD A6          JSR    $FDA6
2374  CE 25 5B          LDX    #LINE7
2377  C6 03             LDAB   #3
2379  BD 2C 17          JSR    PASC
237C  CE 18 E4          LDX    #X
237F  BD 28 1F          JSR    FL2
2382  B6 18 4B          LDAA   SIGN
2385  BD FD 80          JSR    $FD80
2388  86 2E             LDAA   #$2E
238A  BD FD 80          JSR    $FD80
238D  CE 18 06          LDX    #BCD1
2390  BD 2C 02          JSR    WRITE
2393  CE 18 07          LDX    #BCD2
2396  BD 2C 02          JSR    WRITE
2399  CE 25 5E          LDX    #LINE8
239C  C6 04             LDAB   #4
239E  BD 2C 17          JSR    PASC
23A1  7D 18 EB          TST    X+7
23A4  2A 11             BPL    FE30
23A6  4F                CLRA
23A7  5F                CLRB
23A8  F0 18 EB          SUBB   X+7
23AB  82 FF             SBCA   #$FF
23AD  B7 18 02          STAA   TEMP3
23B0  F7 18 03          STAB   TEMP4
23B3  86 2D             LDAA   #$2D
23B5  20 0B             BRA    FE31
23B7  B6 18 EB   FE30   LDAA   X+7
23BA  B7 18 03          STAA   TEMP4
23BD  7F 18 02          CLR    TEMP3
23C0  86 2B             LDAA   #$2B
23C2  BD FD 80   FE31   JSR    $FD80
23C5  BD 2A F0          JSR    BI$BCD
23C8  CE 18 07          LDX    #BCD2
23CB  BD 2C 02          JSR    WRITE
23CE  CE 25 62          LDX    #LINE9
23D1  C6 05             LDAB   #5
23D3  BD 2C 17          JSR    PASC
23D6  CE 18 EC          LDX    #Y
23D9  BD 28 1F          JSR    FL2
23DC  B6 18 4B          LDAA   SIGN
23DF  BD FD 80          JSR    $FD80
23E2  86 2E             LDAA   #$2E
23E4  BD FD 80          JSR    $FD80
23E7  CE 18 06          LDX    #BCD1
23EA  BD 2C 02          JSR    WRITE
23ED  CE 18 07          LDX    #BCD2
23F0  BD 2C 02          JSR    WRITE
23F3  CE 25 5E          LDX    #LINE8
23F6  C6 04             LDAB   #4
23F8  BD 2C 17          JSR    PASC
```

```
23FB  7D 18 F3           TST   Y+7
23FE  2A 11              BPL   FE34
2400  4F                 CLRA
2401  5F                 CLRB
2402  F0 18 F3           SUBB  Y+7
2405  82 FF              SBCA  #$FF
2407  B7 18 02           STAA  TEMP3
240A  F7 18 03           STAB  TEMP4
240D  86 2D              LDAA  #$2D
240F  20 0B              BRA   FE35
2411  B6 18 F3    FE34   LDAA  Y+7
2414  B7 18 03           STAA  TEMP4
2417  7F 18 02           CLR   TEMP3
241A  86 2B              LDAA  #$2B
241C  BD FD 80    FE35   JSR   $FD80
241F  BD 2A F0           JSR   BI$BCD
2422  CE 18 07           LDX   #BCD2
2425  BD 2C 02           JSR   WRITE
2428  BD FD A6           JSR   $FDA6
242B  BD FD A6           JSR   $FDA6
242E  86 07              LDAA  #$07
2430  BD FD 80           JSR   $FD80
2433  CE 25 67           LDX   #LINE10
2436  C6 0A              LDAB  #10D
2438  BD 2C 17           JSR   PASC
243B  BD FD 36           JSR   $FD36
243E  81 59              CMPA  #$59
2440  26 03              BNE   FE36
2442  7E 20 AB           JMP   FE3A
2445  7E FE 2D    FE36   JMP   $FE2D          FANTOM
                  *
                  *
2448  B6 18 33    INCY   LDAA  VJ2
244B  BB 18 42           ADDA  DY+1
244E  19                 DAA
244F  B7 18 33           STAA  VJ2
2452  B6 18 32           LDAA  VJ1
2455  B9 18 41           ADCA  DY
2458  19                 DAA
2459  B7 18 32           STAA  VJ1
245C  B6 18 3C           LDAA  VVJ2
245F  BB 18 44           ADDA  DDY+1
2462  19                 DAA
2463  B7 18 3C           STAA  VVJ2
2466  B6 18 3B           LDAA  VVJ1
2469  B9 18 43           ADCA  DDY
246C  19                 DAA
246D  B7 18 3B           STAA  VVJ1
2470  39                 RTS
                  *
                  *
2471  B6 18 E0    INCXP  LDAA  IPX
2474  F6 18 E1           LDAB  IPX+1
2477  FB 18 48           ADDB  PX+1
247A  B9 18 47           ADCA  PX
247D  B7 18 E0           STAA  IPX
2480  F7 18 E1           STAB  IPX+1
```

```
2483  39                    RTS
                *
                *
2484  B6 18 E2  INCYP       LDAA    IPY
2487  F6 18 E3              LDAB    IPY+1
248A  FB 18 4A              ADDB    PY+1
248D  B9 18 49              ADCA    PY
2490  B7 18 E2              STAA    IPY
2493  F7 18 E3              STAB    IPY+1
2496  39                    RTS
                *
                *
2497            LINE1       EQU     *
2497  50 4F 53              FCC     'POSITION UNIFORM IMAGE.'
24AE  20 54 48              FCC     ' THEN TYPE C '
24BB            LINE2       EQU     *
24BB  43 4F 4D              FCC     'COMPUTES M(P,Q) MOMENTS'
24D2  20 46 52              FCC     ' FROM TRANSFORM'
24E1            LINE3       EQU     *
24E1  53 41 4D              FCC     'SAMPLE AVERAGE IS 2**'
24F6            LINE4       EQU     *
24F6  50 4F 53              FCC     'POSITION TEST IMAGE.'
250A  20 54 48              FCC     ' THEN TYPE M '
2517            LINE5       EQU     *
2517  4D 28 50              FCC     'M(P,Q)'
251D            LINE6       EQU     *
251D  20 20 20              FCC     '      0          1          2'
2533  20 20 20              FCC     '          3          4          5          6          7'
255B            LINE7       EQU     *
255B  58 3D 20              FCC     'X= '
255E            LINE8       EQU     *
255E  20 32 2A              FCC     ' 2**'
2562            LINE9       EQU     *
2562  20 20 59              FCC     '  Y= '
2567            LINE10      EQU     *
2567  43 4F 4E              FCC     'CONTINUE? '
2571            LINE11      EQU     *
2571  4C 49 4D              FCC     'LIMIT='
2577            XZERO       EQU     *
2577  38 57                 FCB     $38,$57
2579            YZERO       EQU     *
2579  34 98                 FCB     $34,$98
257B                        END
```

STATEMENTS =557

FREE BYTES =18705

NO ERRORS DETECTED

MBLE

```
1800                ORG   $1800
1800      TEMP1     RMB   1
1801      TEMP2     RMB   1
1802      TEMP3     RMB   1
1803      TEMP4     RMB   1
1804      TEMP5     RMB   1
1805      TEMP6     RMB   1
1806      BCD1      RMB   1
1807      BCD2      RMB   1
1808      CTR       RMB   1
1809      CTR1      RMB   1
1824                ORG   $1824
1824      COR1      RMB   1
1825      COR2      RMB   1
1826      COR3      RMB   1
1827      COR4      RMB   3
182A      COR9      RMB   1
182B      COR10     RMB   5
1830      UI1       RMB   1
1831      UI2       RMB   1
1832      VJ1       RMB   1
1833      VJ2       RMB   2
1835      DTIME     RMB   1
1836      I         RMB   1
1837      J         RMB   1
1838      K         RMB   1
1839      XST       RMB   2
183B      VVJ1      RMB   1
183C      VVJ2      RMB   1
183D      NSAMP     RMB   1
183E      LOGS      RMB   1
183F      DX        RMB   2
1841      DY        RMB   2
1843      DDY       RMB   2
1845      PO        RMB   2
1847      PX        RMB   2
1849      PY        RMB   2
184B      SIGN      RMB   3
184E      STACK1    RMB   1
184F      STACK2    RMB   1
1850      PUSHST    RMB   1
1851      LIMIT     RMB   1
18A1                ORG   $18A1
18A1      SIGNI     RMB   1
18A2      SIGNQ     RMB   1
18E0                ORG   $18E0
18E0      IPX       RMB   2
18E2      IPY       RMB   2

18E4      X         RMB   8
18EC      Y         RMB   8
18F4      REAL1     RMB   2
18F6      REAL2     RMB   2
18F8      IMAG1     RMB   2
18FA      IMAG2     RMB   2
8000      MATH      EQU   $8000
```

```
1385                ROT15   EQU     $1385
1376                VECT15  EQU     $1376
330F                SYNDET  EQU     $330F
33B8                MUXSEL  EQU     $33B8
2800                        ORG     $2800
                    *       CONVERTS FLPN TO BCD
                    *       X POINTS TO FLPN
                    *       NORMAL ENTRY REQUIRES MAG<1. IF NOT, BCD SET TO 9999
2800  E6 07         FL$BCD  LDAB    7,X             NORMAL ENTRY
2802  2E 57                 BGT     FL5
2804  50                    NEGB
2805  5D            FL1     TSTB
2806  27 17                 BEQ     FL2
2808  A6 00                 LDAA    0,X
280A  47                    ASRA
280B  A7 00                 STAA    0,X
280D  A6 01                 LDAA    1,X
280F  46                    RORA
2810  A7 01                 STAA    1,X
2812  A6 02                 LDAA    2,X
2814  46                    RORA
2815  A7 02                 STAA    2,X
2817  A6 03                 LDAA    3,X
2819  46                    RORA
281A  A7 03                 STAA    3,X
281C  5A                    DECB
281D  20 E6                 BRA     FL1
281F  6D 00         FL2     TST     0,X             FRACTION ONLY
2821  2A 14                 BPL     FL3
2823  4F                    CLRA
2824  BD 2B E5              JSR     PUSH41
2827  4F                    CLRA
2828  BD 2B BF              JSR     PUSH44

282B  86 0C                 LDAA    #12D
282D  BD 80 00              JSR     MATH
2830  86 2D                 LDAA    #$2D
2832  B7 18 4B              STAA    SIGN
2835  20 09                 BRA     FL4
2837  4F            FL3     CLRA
2838  BD 2B BF              JSR     PUSH44
283B  86 2B                 LDAA    #$2B
283D  B7 18 4B              STAA    SIGN
2840  86 DC         FL4     LDAA    #$DC
2842  36                    PSHA
2843  86 46                 LDAA    #$46
2845  36                    PSHA
2846  86 03                 LDAA    #$03
2848  36                    PSHA
2849  4F                    CLRA
284A  36                    PSHA
284B  86 09                 LDAA    #9
284D  BD 80 00              JSR     MATH
2850  CE 18 00              LDX     #TEMP1
2853  4F                    CLRA
2854  BD 2B 99              JSR     PULL4
2857  BD 2A F0              JSR     BI$BCD
```

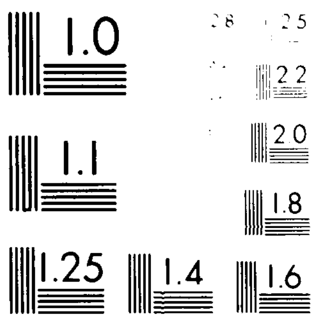
```
285A  39                 RTS
285B  6D 00       FL5    TST    0,X
285D  2A 07              BPL    FL6
285F  86 2D              LDAA   #$2D
2861  B7 18 4B           STAA   SIGN
2864  20 05              BRA    FL7
2866  86 2B       FL6    LDAA   #$2B
2868  B7 18 4B           STAA   SIGN
286B  CE 99 99    FL7    LDX    #$9999
286E  FF 18 06           STX    BCD1
2871  39                 RTS
                  *
                  *
                  *
2872  5D          ACCUM  TSTB
2873  26 03              BNE    AC0
2875  7E 29 55    AC0A   JMP    AC13
2878  F6 18 38    AC0    LDAB   K
287B  C4 07              ANDB   #7
287D  F1 18 51           CMPB   LIMIT
2880  2A F3              BPL    AC0A
2882  F6 18 38           LDAB   K
2885  C4 38              ANDB   #$38
2887  57                 ASRB
2888  57                 ASRB
2889  57                 ASRB
288A  F1 18 51           CMPB   LIMIT
288D  2A E6              BPL    AC0A
288F  FF 18 39           STX    XST
2892  16                 TAB
2893  27 14              BEQ    AC1
2895  5A                 DECB
2896  27 1F              BEQ    AC2
2898  5A                 DECB
2899  27 2A              BEQ    AC3
289B  B6 18 F6           LDAA   REAL2
289E  F6 18 F7           LDAB   REAL2+1
28A1  F0 18 F5           SUBB   REAL1+1
28A4  B2 18 F4           SBCA   REAL1
28A7  20 28              BRA    AC4
28A9  B6 18 F4    AC1    LDAA   REAL1
28AC  F6 18 F5           LDAB   REAL1+1
28AF  FB 18 F7           ADDB   REAL2+1
28B2  B9 18 F6           ADCA   REAL2
28B5  20 1A              BRA    AC4

28B7  B6 18 F8    AC2    LDAA   IMAG1
28BA  F6 18 F9           LDAB   IMAG1+1
28BD  F0 18 FB           SUBB   IMAG2+1
28C0  B2 18 FA           SBCA   IMAG2
28C3  20 0C              BRA    AC4
28C5  B6 18 F8    AC3    LDAA   IMAG1
28C8  F6 18 F9           LDAB   IMAG1+1
28CB  FB 18 FB           ADDB   IMAG2+1
28CE  B9 18 FA           ADCA   IMAG2
28D1  BD 2B 74    AC4    JSR    PUSH82
28D4  B6 18 38           LDAA   K
```

```
28D7   84 07              ANDA    #7
28D9   47                 ASRA
28DA   27 0B              BEQ     AC5
28DC   4A                 DECA
28DD   27 0D              BEQ     AC6
28DF   4A                 DECA
28E0   27 0F              BEQ     AC7
28E2   CE 2D B2           LDX     #C67
28E5   20 0D              BRA     AC8
28E7   CE 2C C2   AC5     LDX     #C01
28EA   20 08              BRA     AC8
28EC   CE 2D 12   AC6     LDX     #C23
28EF   20 03              BRA     AC8
28F1   CE 2D 62   AC7     LDX     #C45
28F4   4F         AC8     CLRA
28F5   F6 18 37           LDAB    J
28F8   58                 ASLB
28F9   49                 ROLA
28FA   BD 2A C8           JSR     IDEX
28FD   A6 00              LDAA    0,X
28FF   E6 01              LDAB    1,X
2901   BD 2B 74           JSR     PUSH82          CJ(Q)
2904   86 01              LDAA    #1
2906   BD 29 EC           JSR     MATH1
2909   B6 18 38           LDAA    K
290C   84 38              ANDA    #$38
290E   47                 ASRA
290F   47                 ASRA
2910   47                 ASRA
2911   47                 ASRA
2912   27 0B              BEQ     AC9
2914   4A                 DECA
2915   27 0D              BEQ     AC10
2917   4A                 DECA
2918   27 0F              BEQ     AC11
291A   CE 2D B2           LDX     #C67
291D   20 0D              BRA     AC12
291F   CE 2C C2   AC9     LDX     #C01
2922   20 08              BRA     AC12
2924   CE 2D 12   AC10    LDX     #C23
2927   20 03              BRA     AC12
2929   CE 2D 62   AC11    LDX     #C45
292C   4F         AC12    CLRA
292D   F6 18 36           LDAB    I
2930   58                 ASLB
2931   49                 ROLA
2932   BD 2A C8           JSR     IDEX
2935   A6 00              LDAA    0,X
2937   E6 01              LDAB    1,X
2939   BD 2B 74           JSR     PUSH82          CI(P)
293C   86 01              LDAA    #1
293E   BD 29 EC           JSR     MATH1
2941   FE 18 39           LDX     XST
2944   BD 2B 5A           JSR     PUSH88          M(P,Q)
2947   86 06              LDAA    #6
2949   BD 80 00           JSR     MATH
294C   FE 18 39           LDX     XST
```

AD-A110 322 DEFT LABS INC EAST SYRACUSE NY F/G 20/1
ACOUSTO-OPTIC TECHNOLOGY FOR TOPOGRAPHIC FEATURE EXTRACTION AND--ETC(U)
MAR 81 A L MOYER DAAK70-79-C-0160
UNCLASSIFIED 0102-A002 ETL-0256 NL

3 of 3
AD A
110322

END
DATE
FILMED
03-82
DTIC


1.0
2.5
2.2
2.0
1.1
1.8
1.25
1.4
1.6

```
294F  BD 2B 40          JSR     PULL8
2952  FE 18 39          LDX     XST
2955  08        AC13    INX
2956  08                INX
2957  08                INX
2958  08                INX
2959  08                INX
295A  08                INX
295B  08                INX
295C  08                INX
295D  7C 18 38          INC     K
2960  39                RTS
                *
                *
                *
2961  BD 2A 96  SAMP    JSR     TUNE
2964  BD 2A DB          JSR     DELAY1
2967  BD 2C 24          JSR     RDDEFT
296A  BD 13 76          JSR     VECT15
296D  4F                CLRA
296E  5F                CLRB
296F  F0 18 2B          SUBB    COR10
2972  B2 18 2A          SBCA    COR9
2975  39                RTS
                *
                *
                *
2976  BD 2A 96  SAMPLE  JSR     TUNE            1ST QUAD
2979  BD 2A DB          JSR     DELAY1
297C  BD 2C 24          JSR     RDDEFT
297F  B6 18 45          LDAA    P0
2982  F6 18 46          LDAB    P0+1
2985  FB 18 E1          ADDB    IPX+1
2988  B9 18 E0          ADCA    IPX
298B  FB 18 E3          ADDB    IPY+1
298E  B9 18 E2          ADCA    IPY
2991  B7 18 2A          STAA    COR9
2994  F7 18 2B          STAB    COR10
2997  BD 13 85          JSR     ROT15
299A  FE 18 24          LDX     COR1
299D  FF 18 F4          STX     REAL1
29A0  FE 18 26          LDX     COR3
29A3  FF 18 F8          STX     IMAG1
29A6  BD 2A AF          JSR     TUNE1           4TH QUAD
29A9  BD 2A DB          JSR     DELAY1
29AC  BD 2C 24          JSR     RDDEFT
29AF  B6 18 45          LDAA    P0
29B2  F6 18 46          LDAB    P0+1
29B5  FB 18 E1          ADDB    IPX+1
29B8  B9 18 E0          ADCA    IPX
29BB  F0 18 E3          SUBB    IPY+1
29BE  B2 18 E2          SBCA    IPY
29C1  B7 18 2A          STAA    COR9
29C4  F7 18 2B          STAB    COR10
29C7  BD 13 85          JSR     ROT15
29CA  FE 18 24          LDX     COR1
29CD  FF 18 F6          STX     REAL2
```

```
29D0  FE 18 26          LDX    COR3
29D3  FF 18 FA          STX    IMAG2
29D6  39                RTS
                  *
                  *
                  *
29D7  B6 18 31   INCX   LDAA   UI2
29DA  BB 18 40          ADDA   DX+1
29DD  19                DAA
29DE  B7 18 31          STAA   UI2
29E1  B6 18 30          LDAA   UI1
29E4  B9 18 3F          ADCA   DX
29E7  19                DAA
29E8  B7 18 30          STAA   UI1
29EB  39                RTS
                  *
                  *
                  *
29EC  33         MATH1  PULB
29ED  F7 18 4E          STAB   STACK1
29F0  33                PULB
29F1  F7 18 4F          STAB   STACK2
29F4  30                TSX
29F5  E6 0F             LDAB   15D,X
29F7  C1 80             CMPB   #$80
29F9  27 05             BEQ    M11
29FB  BD 80 00          JSR    MATH
29FE  20 06             BRA    M13
2A00  86 08      M11    LDAA   $8
2A02  33         M12    PULB
2A03  4A                DECA
2A04  26 FC             BNE    M12
2A06  F6 18 4F   M13    LDAB   STACK2
2A09  37                PSHB
2A0A  F6 18 4E          LDAB   STACK1
2A0D  37                PSHB
2A0E  39                RTS
                  *
                  *
2A0F  6F 00      FZERO  CLR    0,X
2A11  6F 01             CLR    1,X
2A13  6F 02             CLR    2,X
2A15  6F 03             CLR    3,X
2A17  6F 04             CLR    4,X
2A19  6F 05             CLR    5,X
2A1B  6F 06             CLR    6,X
2A1D  C6 80             LDAB   #$80
2A1F  E7 07             STAB   7,X
2A21  08                INX
2A22  08                INX
2A23  08                INX
2A24  08                INX
2A25  08                INX
2A26  08                INX
2A27  08                INX
2A28  08                INX
2A29  39                RTS
```

```
                          *
                          *
                          *      SQUARE ROOT OF FLPTN IN STACK
2A2A   33         SQROOT  PULB
2A2B   F7 18 4E           STAB     STACK1
2A2E   33                 PULB
2A2F   F7 18 4F           STAB     STACK2
2A32   30                 TSX
2A33   C6 18              LDAB     #24D
2A35   09         SQ1     DEX
2A36   5A                 DECB
2A37   26 FC              BNE      SQ1
2A39   FF 18 39           STX      XST
2A3C   C6 08              LDAB     #8D
2A3E   A6 1F      SQ2     LDAA     31D,X
2A40   36                 PSHA
2A41   09                 DEX
2A42   5A                 DECB
2A43   26 F9              BNE      SQ2
2A45   A6 1F              LDAA     31D,X
2A47   47                 ASRA
2A48   A7 1F              STAA     31D,X
2A4A   86 0A              LDAA     #10D
2A4C   B7 18 08           STAA     CTR
2A4F   FE 18 39   SQ3     LDX      XST              MAIN LOOP
2A52   C6 10              LDAB     #16D
2A54   A6 1F      SQ4     LDAA     31D,X
2A56   36                 PSHA
2A57   09                 DEX
2A58   5A                 DECB
2A59   26 F9              BNE      SQ4
2A5B   86 02              LDAA     #2
2A5D   BD 80 00           JSR      MATH             Y=X/IT
2A60   FE 18 39           LDX      XST
2A63   C6 08              LDAB     #8
2A65   A6 17      SQ5     LDAA     23D,X
2A67   36                 PSHA
2A68   09                 DEX
2A69   5A                 DECB
2A6A   26 F9              BNE      SQ5
2A6C   86 06              LDAA     #6
2A6E   BD 80 00           JSR      MATH             Y=Y+IT
2A71   FE 18 39           LDX      XST
2A74   6A 0F              DEC      15D,X
2A76   C6 08              LDAB     #8
2A78   32         SQ6     PULA
2A79   A7 10              STAA     16D,X
2A7B   08                 INX
2A7C   5A                 DECB
2A7D   26 F9              BNE      SQ6
2A7F   7A 18 08           DEC      CTR
2A82   26 CB              BNE      SQ3
2A84   C6 08              LDAB     #8
2A86   32         SQ7     PULA
2A87   A7 10              STAA     16D,X
2A89   08                 INX
2A8A   5A                 DECB
```

```
2A8B  26 F9             BNE     SQ7
2A8D  F6 18 4F          LDAB    STACK2
2A90  37                PSHB
2A91  F6 18 4E          LDAB    STACK1
2A94  37                PSHB
2A95  39                RTS
                *
                *
                *
2A96  B6 18 30  TUNE    LDAA    UI1
2A99  B7 EE 11          STAA    $EE11
2A9C  B6 18 31          LDAA    UI2
2A9F  B7 EE 10          STAA    $EE10
2AA2  B6 18 32          LDAA    VJ1
2AA5  B7 EE 21          STAA    $EE21
2AA8  B6 18 33          LDAA    VJ2
2AAB  B7 EE 20          STAA    $EE20
2AAE  39                RTS
                *
                *
                *
2AAF  B6 18 30  TUNE1   LDAA    UI1
2AB2  B7 EE 11          STAA    $EE11
2AB5  B6 18 31          LDAA    UI2
2AB8  B7 EE 10          STAA    $EE10
2ABB  B6 18 3B          LDAA    VVJ1
2ABE  B7 EE 21          STAA    $EE21
2AC1  B6 18 3C          LDAA    VVJ2
2AC4  B7 EE 20          STAA    $EE20
2AC7  39                RTS
                *
                *
                *
2AC8  FF 18 00  IDEX    STX     TEMP1
2ACB  FB 18 01          ADDB    TEMP2
2ACE  B9 18 00          ADCA    TEMP1
2AD1  B7 18 00          STAA    TEMP1
2AD4  F7 18 01          STAB    TEMP2
2AD7  FE 18 00          LDX     TEMP1
2ADA  39                RTS
                *
                *
                *
2ADB  B6 18 35  DELAY1  LDAA    DTIME
2ADE  4D        D1      TSTA
2ADF  27 05             BEQ     D2
2AE1  8D 04             BSR     DELAY
2AE3  4A                DECA
2AE4  20 F8             BRA     D1
2AE6  39        D2      RTS
                *
                *
                *
2AE7  CE 02 CA  DELAY   LDX     #$2CA
2AEA  09        LDY1    DEX
2AEB  27 02             BEQ     LDY2
2AED  20 FB             BRA     LDY1
```

```
2AEF  39          LDY2    RTS
                  *
                  *
                  *
2AF0  7F 18 06    BI$BCD  CLR     BCD1
2AF3  7F 18 07            CLR     BCD2
2AF6  86 10               LDAA    #16D
2AF8  B7 18 09            STAA    CTR1
2AFB  BF 18 04            STS     TEMP5
2AFE  8E 18 01            LDS     #TEMP3-1
2B01  86 08       LBIB1   LDAA    #8D
2B03  B7 18 08            STAA    CTR
2B06  33                  PULB
2B07  58          LBIB2   ASLB
2B08  24 12               BCC     LBIB3
2B0A  86 01               LDAA    #1
2B0C  BB 18 07            ADDA    BCD2
2B0F  19                  DAA
2B10  B7 18 07            STAA    BCD2
2B13  B6 18 06            LDAA    BCD1
2B16  89 00               ADCA    #0
2B18  19                  DAA
2B19  B7 18 06            STAA    BCD1
2B1C  7A 18 09    LBIB3   DEC     CTR1
2B1F  27 1B               BEQ     LBIB4
2B21  B6 18 07            LDAA    BCD2
2B24  BB 18 07            ADDA    BCD2
2B27  19                  DAA
2B28  B7 18 07            STAA    BCD2
2B2B  B6 18 06            LDAA    BCD1
2B2E  B9 18 06            ADCA    BCD1
2B31  19                  DAA
2B32  B7 18 06            STAA    BCD1
2B35  7A 18 08            DEC     CTR
2B38  27 C7               BEQ     LBIB1
2B3A  20 CB               BRA     LBIB2
2B3C  BE 18 04    LBIB4   LDS     TEMP5
2B3F  39                  RTS
                  *
                  *
                  *
2B40  33          PULL8   PULB
2B41  F7 18 4E            STAB    STACK1
2B44  33                  PULB
2B45  F7 18 4F            STAB    STACK2
2B48  C6 08               LDAB    #8
2B4A  32          P8      PULA
2B4B  A7 00               STAA    0,X
2B4D  08                  INX
2B4E  5A                  DECB
2B4F  26 F9               BNE     P8
2B51  F6 18 4F            LDAB    STACK2
2B54  37                  PSHB
2B55  F6 18 4E            LDAB    STACK1
2B58  37                  PSHB
2B59  39                  RTS
                  *
```

```
                    *
                    *
2B5A  33            PUSH88  PULB
2B5B  F7 18 4E              STAB     STACK1
2B5E  33                    PULB
2B5F  F7 18 4F              STAB     STACK2
2B62  C6 08                 LDAB     #8
2B64  A6 07         P88     LDAA     7,X
2B66  36                    PSHA
2B67  09                    DEX
2B68  5A                    DECB
2B69  26 F9                 BNE      P88
2B6B  F6 18 4F              LDAB     STACK2
2B6E  37                    PSHB
2B6F  F6 18 4E              LDAB     STACK1
2B72  37                    PSHB
2B73  39                    RTS
                    *
                    *
                    *
2B74  F7 18 50      PUSH82  STAB     PUSHST
2B77  33                    PULB
2B78  F7 18 4E              STAB     STACK1
2B7B  33                    PULB
2B7C  F7 18 4F              STAB     STACK2
2B7F  5F                    CLRB
2B80  37                    PSHB
2B81  37                    PSHB
2B82  37                    PSHB
2B83  37                    PSHB
2B84  37                    PSHB
2B85  37                    PSHB
2B86  F6 18 50              LDAB     PUSHST
2B89  37                    PSHB
2B8A  36                    PSHA
2B8B  86 07                 LDAA     #7
2B8D  BD 80 00              JSR      MATH          NFLPTN
2B90  F6 18 4F              LDAB     STACK2
2B93  37                    PSHB
2B94  F6 18 4E              LDAB     STACK1
2B97  37                    PSHB
2B98  39                    RTS
                    *
                    *
                    *
2B99  33            PULL4   PULB
2B9A  F7 18 4E              STAB     STACK1
2B9D  33                    PULB
2B9E  F7 18 4F              STAB     STACK2
2BA1  4D                    TSTA
2BA2  27 06                 BEQ      LP42
2BA4  08            LP41    INX
2BA5  4A                    DECA
2BA6  27 02                 BEQ      LP42
2BA8  20 FA                 BRA      LP41
2BAA  32            LP42    PULA
2BAB  A7 00                 STAA     0,X
```

```
2BAD  32                 PULA
2BAE  A7 01              STAA    1,X
2BB0  32                 PULA
2BB1  A7 02              STAA    2,X
2BB3  32                 PULA
2BB4  A7 03              STAA    3,X
2BB6  F6 18 4F           LDAB    STACK2
2BB9  37                 PSHB
2BBA  F6 18 4E           LDAB    STACK1
2BBD  37                 PSHB
2BBE  39                 RTS
                *
                *
                *
2BBF  33        PUSH44   PULB
2BC0  F7 18 4E           STAB    STACK1
2BC3  33                 PULB
2BC4  F7 18 4F           STAB    STACK2
2BC7  4D                 TSTA
2BC8  27 06              BEQ     LP442
2BCA  08        LP441    INX
2BCB  4A                 DECA
2BCC  27 02              BEQ     LP442
2BCE  20 FA              BRA     LP441
2BD0  A6 03     LP442    LDAA    3,X
2BD2  36                 PSHA
2BD3  A6 02              LDAA    2,X
2BD5  36                 PSHA
2BD6  A6 01              LDAA    1,X
2BD8  36                 PSHA
2BD9  A6 00              LDAA    0,X
2BDB  36                 PSHA
2BDC  F6 18 4F           LDAB    STACK2
2BDF  37                 PSHB
2BE0  F6 18 4E           LDAB    STACK1
2BE3  37                 PSHB
2BE4  39                 RTS
                *
                *
                *
2BE5  33        PUSH41   PULB
2BE6  F7 18 4E           STAB    STACK1
2BE9  33                 PULB
2BEA  F7 18 4F           STAB    STACK2
2BED  36                 PSHA
2BEE  48                 ASLA
2BEF  24 04              BCC     LP411
2BF1  86 FF              LDAA    #$FF
2BF3  20 01              BRA     LP412
2BF5  4F        LP411    CLRA
2BF6  36        LP412    PSHA
2BF7  36                 PSHA
2BF8  36                 PSHA
2BF9  F6 18 4F           LDAB    STACK2
2BFC  37                 PSHB
2BFD  F6 18 4E           LDAB    STACK1
2C00  37                 PSHB
```

```
2C62  A7 27              STAA   $27,X
2C64  6A 08              DEC    8,X
2C66  20 CE              BRA    RD1
2C68  A6 3E     RD2      LDAA   $3E,X
2C6A  A7 08              STAA   8,X
2C6C  6D 08     RD3      TST    8,X
2C6E  27 22              BEQ    RD4
2C70  A6 24              LDAA   $24,X
2C72  47                 ASRA
2C73  A7 24              STAA   $24,X
2C75  A6 25              LDAA   $25,X
2C77  46                 RORA
2C78  A7 25              STAA   $25,X
2C7A  A6 26              LDAA   $26,X
2C7C  46                 RORA
2C7D  A7 26              STAA   $26,X
2C7F  A6 27              LDAA   $27,X
2C81  47                 ASRA
2C82  A7 27              STAA   $27,X
2C84  A6 28              LDAA   $28,X
2C86  46                 RORA
2C87  A7 28              STAA   $28,X
2C89  A6 29              LDAA   $29,X
2C8B  46                 RORA
2C8C  A7 29              STAA   $29,X
2C8E  6A 08              DEC    8,X
2C90  20 DA              BRA    RD3
2C92  FE 18 25  RD4      LDX    COR2
2C95  FF 18 24           STX    COR1
2C98  FE 18 28           LDX    COR4+1
2C9B  FF 18 26           STX    COR3
2C9E  39                 RTS
                *
                *
                *
2C9F  86 03     READSN   LDAA   #3
2CA1  BD 33 B8           JSR    MUXSEL
2CA4  CE 18 00           LDX    #TEMP1
2CA7  BD 33 0F           JSR    SYNDET
2CAA  7D 18 00           TST    TEMP1
2CAD  8D 0C              BSR    SGNSET
2CAF  B7 18 A1           STAA   SIGNI
2CB2  7D 18 02           TST    TEMP3
2CB5  8D 04              BSR    SGNSET
2CB7  B7 18 A2           STAA   SIGNQ
2CBA  39                 RTS
                *
                *
                *
2CBB  2A 03     SGNSET   BPL    SETPOS
2CBD  86 FF              LDAA   #$FF
2CBF  39                 RTS
2CC0  4F        SETPOS   CLRA
2CC1  39                 RTS
                *
                *
                *
```

```
2CC2                C01     EQU     *
2CC2   40 00 98             FCB     $40,0,$98,$33
2CC6   00 00 0B             FCB     0,0,$B,$95
2CCA   00 00 FB             FCB     0,0,$FB,$CD
2CCE   00 00 02             FCB     0,0,2,$2C
2CD2   00 00 FE             FCB     0,0,$FE,$AB
2CD6   00 00 00             FCB     0,0,0,$EA
2CDA   00 00 FF             FCB     0,0,$FF,$55
2CDE   00 00 00             FCB     0,0,0,$85
2CE2   00 00 FF             FCB     0,0,$FF,$96
2CE6   00 00 00             FCB     0,0,0,$59
2CEA   00 00 FF             FCB     0,0,$FF,$B5
2CEE   00 00 00             FCB     0,0,0,$42
2CF2   00 00 FF             FCB     0,0,$FF,$C6
2CF6   00 00 00             FCB     0,0,0,$36
2CFA   00 00 FF             FCB     0,0,$FF,$CF
2CFE   00 00 00             FCB     0,0,0,$3D
2D02   00 00 FF             FCB     0,0,$FF,$D5
2D06   00 00 00             FCB     0,0,0,$2B
2D0A   00 00 FF             FCB     0,0,$FF,$D7
2D0E   00 00 00             FCB     0,0,0,$29
2D12                C23     EQU     *
2D12   40 00 90             FCB     $40,0,$90,$44
2D16   BD F4 1D             FCB     $BD,$F4,$1D,$E
2D1A   00 00 0A             FCB     0,0,$A,$95
2D1E   02 72 FC             FCB     2,$72,$FC,$51
2D22   00 00 FE             FCB     0,0,$FE,$9F
2D26   FF 7A 00             FCB     $FF,$7A,0,$C5
2D2A   00 00 00             FCB     0,0,0,$8B
2D2E   00 31 FF             FCB     0,$31,$FF,$A8
2D32   00 00 FF             FCB     0,0,$FF,$CB
2D36   FF EA 00             FCB     $FF,$EA,0,$27
2D3A   00 00 00             FCB     0,0,0,$1F
2D3E   00 0C FF             FCB     0,$C,$FF,$EA
2D42   00 00 FF             FCB     0,0,$FF,$F1
2D46   FF FA 00             FCB     $FF,$FA,0,$C
2D4A   00 00 00             FCB     0,0,0,$A
2D4E   00 04 FF             FCB     0,4,$FF,$F9
2D52   00 00 FF             FCB     0,0,$FF,$FC
2D56   FF FF 00             FCB     $FF,$FF,0,3
2D5A   00 00 00             FCB     0,0,0,2
2D5E   00 01 00             FCB     0,1,0,0
2D62                C45     EQU     *
2D62   40 00 8E             FCB     $40,0,$8E,$71
2D66   B4 EF 21             FCB     $B4,$EF,$21,$B5
2D6A   00 00 10             FCB     0,0,$10,$2A
2D6E   0D 5A F7             FCB     $D,$5A,$F7,$8E
2D72   00 00 FB             FCB     0,0,$FB,$70
2D76   FC F1 02             FCB     $FC,$F1,2,$9B
2D7A   00 00 01             FCB     0,0,1,$96
2D7E   01 21 FE             FCB     1,$21,$FE,$F5
2D82   00 00 FF             FCB     0,0,$FF,$47
2D86   FF 7A 00             FCB     $FF,$7A,0,$84
2D8A   00 00 00             FCB     0,0,0,$61
2D8E   00 48 FF             FCB     0,$48,$FF,$B8
2D92   00 00 FF             FCB     0,0,$FF,$CA
2D96   FF D8 00             FCB     $FF,$D8,0,$2A
```

```
2D9A  00 00 00          FCB   0,0,0,$20
2D9E  00 18 FF          FCB   0,$18,$FF,$E9
2DA2  00 00 FF          FCB   0,0,$FF,$F0
2DA6  FF F5 00          FCB   $FF,$F5,0,$C
2DAA  00 00 00          FCB   0,0,0,7
2DAE  00 04 FF          FCB   0,4,$FF,$FF
2DB2           C67      EQU   *
2DB2  40 00 8D          FCB   $40,0,$8D,$BF
2DB6  B2 03 23          FCB   $B2,3,$23,$98
2DBA  00 00 12          FCB   0,0,$12,$BC
2DBE  12 A5 F4          FCB   $12,$A5,$F4,$E2
2DC2  00 00 F9          FCB   0,0,$F9,$18
2DC6  F9 95 04          FCB   $F9,$95,4,$6B
2DCA  00 00 02          FCB   0,0,2,$E9
2DCE  02 96 FE          FCB   2,$96,$FE,$5
2DD2  00 00 FE          FCB   0,0,$FE,$9B
2DD6  FE C1 01          FCB   $FE,$C1,1,4
2DDA  00 00 00          FCB   0,0,0,$C1
2DDE  00 AE FF          FCB   0,$AE,$FF,$EF
2DE2  00 00 FF          FCB   0,0,$FF,$92
2DE6  FF 9D 00          FCB   $FF,$9D,0,$55
2DEA  00 00 00          FCB   0,0,0,$41
2DEE  00 3A FF          FCB   0,$3A,$FF,$D0
2DF2  00 00 FF          FCB   0,0,$FF,$DD
2DF6  FF E2 00          FCB   $FF,$E2,0,$18
2DFA  00 00 00          FCB   0,0,0,$E
2DFE  00 0A FF          FCB   0,$A,$FF,$FC
2E02  CE 40 00 DUMMY    LDX   #$4000
2E05  FF 18 24          STX   COR1
2E08  CE 00 00          LDX   #0
2E0B  FF 18 26          STX   COR3
2E0E  39                RTS
2E0F                    END
```

STATEMENTS =832

FREE BYTES =16585

NO ERRORS DETECTED

APPENDIX C - Fortran Code Listings

This Appendix consists of listings of Fortran programs which can be used to compute the coefficients $c_k(p)$ which are defined by equations (104), (105) and (106). These coefficients are used to compute image moments from the image Fourier transform. The program PEVEN computes the coefficients for the case that p is even. The program PODD computes the coefficients for the case that p is odd.

This Appendix also contains listings of the programs EVENER and ODDER. These programs were used to obtain the data which is presented in Figure 26 and Tables 3 and 4. EVENER evaluates equation (137) for the case of p even. ODDER evaluates the same equation for the case of p odd.

```
/      PEVEN
       PI=3.1415927
       TYPE 10
10     FORMAT(' COMPUTES CK(P); P EVEN')
       TYPE 20
20     FORMAT('  ')
       TYPE 30
30     FORMAT(' INPUT N/2, P(MAX)')
       TYPE 20
       ACCEPT 40,N
       ACCEPT 40,NPM
40     FORMAT(I3)
       N=N+N
       TYPE 20
       TYPE 50
50     FORMAT(' COEFFICIENTS  (REAL)')
       TYPE 20
       TYPE 60
60     FORMAT('  K    P       CK(P)     OCTAL')
       DO 105 NP=0,NPM,2
       DO 100 K=0,N,1
       C=0.
       IF(NP.EQ.0) C=.5
       DO 90 L=1,N,1
       X=2.*FLOAT(L)/FLOAT(N)
       IF(X.GT.1.) GO TO 70
       FX=X**FLOAT(NP)
       GO TO 80
70     FX=2-(X-2)**FLOAT(NP)
80     ARG=FLOAT(K)*PI*X/2.
       IF(L.EQ.N) C=C-.5*FX*COS(ARG)
90     C=C+FX*COS(ARG)
       C=C/FLOAT(N)
       NC=.5+32768.*C
       IF(NC.LT.0) NC=65536+NC
100    TYPE 110,K,NP,C,NC
105    TYPE 20
110    FORMAT(I3,I5,'   ',F11.7,'   ',O6)
       STOP
       END
**
```

```
      /    PODD
           PI=3.1415927
           TYPE 10
   10      FORMAT(' COMPUTES CK(P); P ODD')
           TYPE 20
   20      FORMAT(' ')
           TYPE 30
   30      FORMAT(' INPUT N/4, P(MAX)')
           TYPE 20
           ACCEPT 40,N
           ACCEPT 40,NPM
   40      FORMAT (I3)
           N=N+N+N+N
           TYPE 20
           TYPE 50
   50      FORMAT(' COEFFICIENTS  (IMAGINARY)')
           TYPE 20
           TYPE 60
   60      FORMAT('  K    P       CK(P)     OCTAL')
           DO 140 NP=1,NPM,2
           DO 130 K=1,N,1
           C=0.
           DO 120 L=1,N,1
           X=4.*FLOAT(L)/FLOAT(N)
           IF(X.GT.1.) GO TO 70
           FX=X**FLOAT(NP)
           GO TO 110
   70      IF(X.GT.2.) GO TO 90
   80      FX=2.+(X-2.)**FLOAT(NP)
           GO TO 110
   90      IF(X.GT.3.) GO TO 100
           FX=2.-(X-2.)**FLOAT(NP)
           GO TO 110
   100     FX=-(X-4.)**FLOAT(NP)
   110     ARG=FLOAT(K)*PI*X/4.
           IF(L.EQ.N) C=C-.5*FX*SIN(ARG)
   120     C=C+FX*SIN(ARG)
           C=-C/FLOAT(N)
           NC=.5+32768.*C
           IF(NC.LT.0) NC=65536+NC
   130     TYPE 150, K,NP,C,NC
   140     TYPE 20
   150     FORMAT (I3,I5,'   ',F11.7,'   ',O6)
           STOP
           END
   **
```

```
/       EVENER

        DIMENSION CFF(50)
        PI=3.1415927
        TYPE 10
 10     FORMAT(' COMPUTES ER(P,X); P EVEN')
 15     TYPE 20
 20     FORMAT('  ')
        TYPE 30
 30     FORMAT(' INPUT N/2, P')
        TYPE 20
        ACCEPT 40,N
        ACCEPT 40,NP
 40     FORMAT(I3)
        N=N+N
        TYPE 20
        DO 100 K=0,N,1
        C=0.
        IF(NP.EQ.0) C=.5
        DO 90 L=1,N,1
        X=2.*FLOAT(L)/FLOAT(N)
        IF(X.GT.1.) GO TO 70
        FX=X**FLOAT(NP)
        GO TO 80
 70     FX=2-(X-2)**FLOAT(NP)
 80     ARG=FLOAT(K)*PI*X/2.
        IF(L.EQ.N) C=C-.5*FX*COS(ARG)
 90     C=C+FX*COS(ARG)
        C=C/FLOAT(N)
 100    CFF(K+1)=C
        TYPE 110
 110    FORMAT(' INPUT MESH DIVISOR=M')
        TYPE 20
        ACCEPT 40,M
        TYPE 20
        TYPE 120
 120    FORMAT (' Q        ER(P,2*Q/MN)')
        NQS=.5*FLOAT(N*M)
        DO 140 NQ=0,NQS,1
        X=FLOAT(2*NQ)/FLOAT(M*N)
        ARG=PI*X/2.
        ER=0.
        DO 130 K=1,N,1
        ARGK=ARG*FLOAT(K)
 130    ER=ER-CFF(K+1)*COS(ARGK)
        ER=ER-.5*(CFF(1)-CFF(N+1)*COS(ARGK))
        ER=2.*ER+X**FLOAT(NP)
 140    TYPE 150,NQ,ER
 150    FORMAT(I4,'    ',F11.7)
        TYPE 160
 160    FORMAT (' TO CONTINUE TYPE 1')
        TYPE 20
        ACCEPT 40,N
        IF(N-1)170,15,170
 170    STOP
        END
 **
```

```
/       ODDER
        DIMENSION CFF(50)
        PI=3.1415927
        TYPE 10
  10    FORMAT(' COMPUTES ER(P,X); P ODD')
  15    TYPE 20
  20    FORMAT(' ')
        TYPE 30
  30    FORMAT(' INPUT N/4, P')
        TYPE 20
        ACCEPT 40,N
        ACCEPT 40,NP
  40    FORMAT (I3)
        N=N+N+N+N
        TYPE 20
        DO 130 K=1,N,1
        C=0.
        DO 120 L=1,N,1
        X=4.*FLOAT(L)/FLOAT(N)
        IF(X.GT.1.) GO TO 70
        FX=X**FLOAT(NP)
        GO TO 110
  70    IF(X.GT.2.) GO TO 90
  80    FX=2.+(X-2.)**FLOAT(NP)
        GO TO 110
  90    IF(X.GT.3.) GO TO 100
        FX=2.-(X-2.)**FLOAT(NP)
        GO TO 110
  100   FX=-(X-4.)**FLOAT(NP)
  110   ARG=FLOAT(K)*PI*X/4.
        IF(L.EQ.N) C=C-.5*FX*SIN(ARG)
  120   C=C+FX*SIN(ARG)
        C=C/FLOAT(N)
  130   CFF(K+1)=C
        TYPE 140
  140   FORMAT(' INPUT MESH DIVISOR=M')
        TYPE 20
        ACCEPT 40,M
        TYPE 20
        TYPE 150
  150   FORMAT(' Q        ER(P,2*Q/MN)')
        NQS=.25*FLOAT(N*M)
        DO 170 NQ=0,NQS,1
        X=FLOAT(4*NQ)/FLOAT(M*N)
        ARG=PI*X/4.
        ER=0.
        DO 160 K=1,N,1
        ARGK=ARG*FLOAT(K)
  160   ER=ER-CFF(K+1)*SIN(ARGK)
        ER=2.*ER+X**FLOAT(NP)
  170   TYPE 180,NQ,ER
  180   FORMAT(I4,'   ',F11.7)
        TYPE 190
  190   FORMAT(' TO CONTINUE TYPE 1')
        TYPE 20
        ACCEPT 40,N
        IF(N-1) 200,15,200
  200   STOP
        END
```

Distribution:

W26HAJ U.S. Army Engineer Topographic Laboratories Research Institute Fort Belvoir, VA 22060 Attn: Joseph F. Hannigan	2 copies
W26HAJ/STINFO U.S. Army Engineer Topographic Laboratories Attn: ETL - STINFO Fort Belvoir, VA 22060	14 copies

END

DATE
FILMED

03-82

DTIC